34390

National Library of Canada / Bibliothèque nationale du Canada

CANADIAN THESES ON MICROFICHE / *THÈSES CANADIENNES SUR MICROFICHE*

NAME OF AUTHOR/*NOM DE L'AUTEUR* SURENDER KUMAR KENUE

TITLE OF THESIS/*TITRE DE LA THÈSE* ON THE DETECTION OF CHANGES IN DIGITAL IMAGES

UNIVERSITY/*UNIVERSITÉ* UNIV. OF ALBERTA

DEGREE FOR WHICH THESIS WAS PRESENTED/ *GRADE POUR LEQUEL CETTE THÈSE FUT PRÉSENTÉE* DOCTOR OF PHILOSPHY

YEAR THIS DEGREE CONFERRED/*ANNÉE D'OBTENTION DE CE GRADE* FALL 1977

NAME OF SUPERVISOR/*NOM DU DIRECTEUR DE THÈSE* PROF. WAYNE A. DAVIS

Permission is hereby granted to the NATIONAL LIBRARY OF CANADA to microfilm this thesis and to lend or sell copies of the film.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

L'autorisation est, par la présente, accordée à la BIBLIOTHÈQUE NATIONALE DU CANADA de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans l'autorisation écrite de l'auteur.

DATED/*DATÉ* Sept 12, 77 SIGNED/*SIGNÉ* Surender Kr Kenue

PERMANENT ADDRESS/*RÉSIDENCE FIXÉ* 1213, SECTOR: IV
R.K. PURAM
NEW DELHI - 110022 INDIA

NL-91 (3-74)

National Library of Canada

Cataloguing Branch
Canadian Theses Division

Ottawa, Canada
K1A 0N4

Bibliothèque nationale du Canada

Direction du catalogage
Division des thèses canadiennes

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us a poor photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

THIS DISSERTATION
HAS BEEN MICROFILMED
EXACTLY AS RECEIVED

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de mauvaise qualité.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30. Veuillez prendre connaissance des formules d'autorisation qui accompagnent cette thèse.

LA THÈSE A ÉTÉ
MICROFILMÉE TELLE QUE
NOUS L'AVONS REÇUE

THE UNIVERSITY OF ALBERTA

ON THE DETECTION OF CHANGES IN DIGITAL IMAGES

by

© SURENDER KUMAR KENUE

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF DOCTOR OF PHILOSOPHY

IN

COMPUTING SCIENCE

DEPARTMENT OF COMPUTING SCIENCE

EDMONTON, ALBERTA

FALL 1977

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled 'On the detection of changes in digital images', submitted by Surender Kumar Kenue in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Computing Science.

Wayne A. Davis
Supervisor

A. Calay

P. H. Enslow

K. Schubert

J. G. Peet
External Examiner

DATE 12 Sept 77

ABSTRACT

This thesis deals with the detection of changes in digital pictures. If pictures have been obtained at different times and under different environmental conditions, two basic steps, namely registration and normalization, are essential for change detection. A normalization method is developed which changes the histogram distribution of a picture to any desired distribution. Also, a picture enhancement method is proposed for enhancing small features, which may not be ordinarily visible. Applications of this method include improved edge and contour detection and noise removal.

Registration is concerned with the precise alignment of two or more pictures. An automatic selection procedure for control points is given and some problems encountered in selecting them are presented. If the coordinates of the control points are incorrect, an iterative feedback approach is presented to correct them. The procedure was applied to a number of pictures of sizes 256*256 to 501*501, with a resulting registration accuracy of about 1 to 2 pixels for the first order transformation and about 1 pixel for the second order transformation. The registered pictures were generated by the nearest neighbor, bilinear interpolation, and cubic convolution techniques. A technique for the registration of pictures with large scale differences is

given. For template matching, a fast algorithm which uses heuristic information is developed and compared with other methods.

The detection of changes is accomplished by subtracting gray levels in the corresponding pixels of the registered pictures. The difference pictures are then thresholded to yield binary change pictures, in which isolated change pixels are removed.

ACKNOWLEDGEMENTS

I would like to thank Prof. W.A. Davis, my supervisor, for his advice, guidance and criticism in this research. This thesis could not have been completed without his help.

I would also like to thank Dr. Len Schubert and Dr. S. Cabay for their advice and help during various stages of the thesis. Thanks are also extended to Dr. F.G. Peet and Dr. P.H. Crown for their valuable suggestions.

Finally, financial support in terms of teaching and research assistantships is gratefully acknowledged from the Department of Computing Science, and the National Research Council under grant NRC A7634.

TABLE OF CONTENTS

CHAPTER		PAGE
I.	INTRODUCTION	1
	1.1 Introduction	2
	1.2 Notation and Basic Definitions	4
	1.3 Organization of Thesis	9
II.	PREPROCESSING OF PICTURES	11
	2.1 Gray Scale Normalization	11
	2.1.1 Arbitrary Histogram Shape Method	13
	2.2 Picture Enhancement	22
	2.2.1 The Proposed Method	23
	2.3 Conclusions and Test Results	26
	2.3.1 Gray Scale Normalization	30
	2.3.2 Picture Enhancement	35
III.	REGISTRATION	42
	3.1 Introduction	43
	3.2 Survey of Registration Methods	45
	3.3 Automatic Selection of Control Points	47
	3.3.1 Features of a Good Control Point	49
	3.3.2 Spatial Distribution of Control Points	52
	3.4 Iterative Feedback Approach	63
	3.5 Registration with Large Scale Differences	76
	3.6 Discussion of Results	85
IV.	TEMPLATE MATCHING	87
	4.1 The Proposed Method	89

4.2 Fast Correlation Coefficient 96

4.2.1 Algorithm E for Sum x and Sum x**2 98

4.2.2 Algorithm F for Sum x and Sum x**2 100

4.3 Discussion of Results 105

V. CHANGE DETECTION 106

5.1 Survey 107

5.2 Type of Changes 110

5.3 Method of Solution 111

5.4 Discussion 132

VI. CONCLUSIONS 133

REFERENCES 138

APPENDIX A - List of Software Programs 147

APPENDIX B - Summary of Picture Details 149

LIST OF TABLES

Table	Description	Page
I	A Picture Matrix.	5
II	Histogram of Picture in Table I.	6
III	Binary Picture from Table I with Threshold 10.	6
IV	Normalized Picture Matrix of Table I.	7
V	Decomposition Matrix.	20
VI	Original Matrix.	25
VII	Binary Neighborhood Matrix.	26
VIII	Enhanced Matrix.	26
IX	Degree of Connectedness for Spinach.	29
X	Degree of Connectedness for Neuron.	29
XI	Spatial Distribution of 64 Points.	52
XII	Mapping of Control Point Numbers.	53
XIII	Computational Times for Correlation Methods I.	93
XIV	Computational Times for Correlation Methods II.	94
XV	Number of Muliplications and Additions for Matching Methods.	104

XVI	Original Matrix A.	115
XVII	Registered Matrix B.	115
XVIII	Binary Change Matrix C.	115
XIX	Two Dimensional Histograms.	115
XX	Means of Change and No-Change Gray Levels.	120

LIST OF FIGURES

Figures		Page
1.	Neuron.	16
2.	Neuron with Gaussian histogram.	16
3.	Histogram of Neuron.	17
4.	Normalized Gaussian histogram.	18
5.	Neuron with flat histogram.	19
6.	Spinach.	28
7.	Histogram of Spinach.	28
8.	Spinach with flat histogram.	31
9.	Spinach with Gaussian histogram.	31
10.	Edges of Neuron.	32
11.	Edges of Neuron with flat histogram.	32
12.	Edges of Neuron with Gaussian histogram.	33
13.	Edges of Spinach.	33
14.	Edges of Spinach with flat histogram.	34
15.	Edges of Spinach with Gaussian histogram.	34
16.	Kramer Neuron.	36
17.	Kramer Spinach.	36
18.	Weszka Neuron.	37
19.	Weszka Spinach.	37
20.	Edges of Weszka Neuron.	38
21.	Edges of Weszka Spinach.	38
22.	Enhanced Neuron.	40
23.	Enhanced Spinach.	40

LIST OF FIGURES (continued)

24. Edges of enhanced Neuron. 41

25. Edges of enhanced Spinach. 41

26. Porty nine potential control points. 54

27. Control point numbers. 54

28. Edmonton I with control points. 58

29. Edmonton II with control points. 58

30. Binary gradient picture of Edmonton I. 60

31. Fertilizer 2 with control points. 62

32. Fertilizer 3 with control points. 62

33. Lake I. 67

34. Lake II. 67

35. Lake II registered with Lake I using nearest neighbor approximation. 69

36. Fertilizer 3 registered with Fertilizer 2 using nearest neighbor approximation. 69

37. Fertilizer 3 registered with Fertilizer 2 using bilinear approximation. 72

38. Fertilizer 3 registered with Fertilizer 2 using cubic convolution. 72

39. Edmonton I registered with Edmonton II using nearest neighbor approximation. 73

40. Forestry Cut I. 75

41. Forestry Cut II. 75

42. Forestry Cut I registered with Forestry Cut II using nearest neighbor approximation. 75

LIST OF FIGURES (continued)

43. Rectangular region. 76

44. Zeroth order approximation to the rectangular region. 79

45. Window of Edmonton II. 82

46. 4*4 average of aerial picture. 82

47. Window of aerial picture. 83

48. Average aerial picture registered with Edmonton II using nearest neighbor approximation. 83

49. Aerial picture registered with Edmonton II using nearest neighbor approximation. 84

50. Aerial picture registered with Edmonton II using proposed approximation. 84

51. Large and small pictures. 97

52. Computation of running sums. 98

53. Lake I with Gaussian histogram. 117

54. Lake II (Gaussian) registered with Lake I (Gaussian) using nearest neighbor approximation. 117

55. Histogram of difference picture of Lake area. 118

56. Binary change picture of Lake area. 119

57. Binary change picture of Edmonton using bands 4 and 5. 119

58. Binary change picture of Edmonton using

LIST OF FIGURES (continued)

bands 6 and 7. 122

59. Overlaid change picture of Edmonton II using bands 4 and 5. 122

60. Overlaid change picture of Edmonton II using bands 6 and 7. 123

61. Window of Edmonton II with Gaussian histogram. 123

62. Aerial picture (Gaussian) registered with Edmonton II (Gaussian) using proposed approximation. 125

63. Binary change picture of aerial and Edmonton II. 125

64. Forestry Cut I (Gaussian) registered with Forestry Cut II (Gaussian) using nearest neighbor approximation. 126

65. Forestry Cut II with Gaussian histogram. 126

66. Binary change picture of cut area. 126

67. Fertilizer 2 with Gaussian histogram. 128

68. Fertilizer 3 (Gaussian) registered with Fertilizer 2 (Gaussian) using nearest neighbor approximation. 128

69. Fertilizer 3 (Gaussian) registered with Fertilizer 2 (Gaussian) using bilinear approximation. 129

LIST OF FIGURES (continued)

70. Fertilizer 3 (Gaussian) registered with Fertilizer 2 (Gaussian) using cubic convolution. 129

71. Binary change picture of Fertilizer area using nearest neighbor approximation. 130

72. Binary change picture of Fertilizer area using bilinear approximation. 130

73. Binary change picture of Fertilizer area using cubic convolution. 131

Chapter I

INTRODUCTION

This thesis is concerned with detecting differences between two pictures of the same scene taken at different times or alignments. In a manual change detection system [36], an image interpreter would mount the two photographs, align one with the other and then obtain a 'difference picture'. On the other hand, an automatic system should take a series of input pictures, process them, and then output any change observed. The problem of change detection can be separated into two basic steps, firstly arranging the two pictures by suitable transformations so that they are properly aligned with each other, and then detecting various types of changes by some suitable set of operators.

Applications of change detection include military reconnaissance, city planning, plant science, forestry, pollution control, etc. For example, it is possible to detect the growth or defoliation of plants from two pictures. Further examples of specific change detection systems are briefly summarized in Chapter V. In this thesis change detection is examined in detail and a methodology for automatic change detection is proposed.

Since a LANDSAT satellite provides repetitive images

of the same scene every 18 days, the development of an automatic system is justified. Such a system could monitor seasonal changes in the same year, as well as changes between the same seasons in different years. The monitoring of changes are of immense use to resource managers in the areas of hydrology, pedology, and forestry. Also, in a real time environment, i.e., continuous pictures instead of just still pictures, the application of an automatic system is important. For example, in military applications, it is possible to monitor when a missile shows up on its launching pad, and in forestry it is possible to check for the occurrence of fires. A survey of applications of picture processing to remote sensing can be found in [10,40].

The processing of pictures can be done either optically or digitally. Optical techniques are fast and work well on smaller pictures; however, applying non-linear operations is not always easy. Digital techniques offer the advantage that the original data can be transmitted and manipulated without degradation and are relatively easier for applying non-linear operations. This thesis will restrict itself to digital techniques. The following paragraphs will give a brief introduction to digital picture processing.

1.1 Introduction

Digital picture processing is concerned with manipulation of digital pictures for various types of

applications in applied sciences. It deals with topics such as: enhancement, restoration, coding and analysis. The purpose of enhancement [20,32,51] is to smooth or sharpen pictures for either visual inspection or for further processing by machine. Smoothing averages noise and may blur the picture, while sharpening deblurs the picture and permits object boundaries to be easily identified.

Restoration [3,26,57] consists of transforming a degraded picture to an approximation of the original picture. The degradation can occur due to motion blur, defocusing, etc., and it is generally assumed that some knowledge about the degradation process is available.

Picture coding [1,19,65] refers to the management and manipulation of data either to minimize storage or to maximize the transmission rate. In one approach, statistical features from blocks of data are stored or transmitted and the picture can be recovered from these measurements. In another approach, using derivative type operations, the low and high frequency parts of a picture can be retained and again the picture can be reconstructed. Fourier, Hadamard and other types of transforms can also be used for efficient data compression.

Finally, the problem of extracting information such as statistical and textural measures, parts of pictures which match, object recognition, pattern classification, etc., are all part of picture analysis. Template matching finds its application in object recognition, and for locating control

points in image registration. Pattern classification techniques deal with classifying a picture into various regions of similar properties. For example, crops, water, marsh lands, and forests can easily be classified into their various catagories. A review of recent techniques can be found in the excellent survey papers [48-50].

1.2 Notation and Basic Definitions

In this section, the notation used and some basic definitions are given.

The operator '*' will be used as a multiplication sign, e.g., in y = a*x + b, a is multiplied by x and then added to b.

The symbol 'K' will represent the expression: 2**b - 1, where '**' is the exponentiation operator.

The symbol 'E' will represent the expression: 10**, where '**' is the exponentiation operator.

The symbol '<--' will represent the assignment operator, e.g., x <-- y, means that the value of variable x is replaced by the value of variable y.

The symbol 'Sum' will be used in place of the capital sigma of the Greek alphabet, e.g., Sum x*y means the inner product sum of vectors x and y. The range of the vectors and the Sum is from 1 to n, for n-tuple vectors x and y.

The symbol 'Sqrt( )' refers to the square root of the expression inside the parenthesis, e.g., Sqrt(x + y) is the square root of x + y.

The symbol 'abs( )' refers to the absolute value of the expression inside the parenthesis, e.g., abs(x + y) is the absolute value of the expression x + y.

A picture function P is a continuous function of two variables defined on some bounded region of a plane.

A digital picture function is a discrete function of two variables, in which the total sampling area is divided into picture elements or pixels. The term 'picture' or 'image' will be used to mean a digital picture function.

A gray level (or gray value) is an integer value assigned to every pixel of the picture, and is proportional to the relative amounts of energy coming from each instantaneous field of view within the entire scene. The gray level range for a b-bit pixel, is from 0 to K, 0 being the maximum black value, and K the maximum white value, where b represents the number of bits per pixel.

A histogram H is a vector of frequencies of gray levels. For example, the histogram of the 6*6 picture matrix in Table I is given in Table II.

Table I. A Picture Matrix.

3	7	7	7	7	10
4	6	6	6	7	12
5	5	5	12	12	12
4	4	4	12	13	15
5	5	6	12	15	15
10	10	11	11	12	13

Table II. Histogram of Picture in Table I.

Frequency	0	0	0	1	4	5	4	5	0	0	3	2	7	2	0	3
Gray Level	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

A binary thresholding refers to the threshold T such that all gray levels between 0 and T and from T + 1 to K, (for a b-bit pixel) are mapped to 0 and 1 respectively. When this transformation is applied to the original picture, a binary picture is obtained. For example, if the threshold T=10 is applied to the picture in Table I, the binary picture given in Table III results.

Table III. Binary Picture from Table I with Threshold 10.

0	0	0	0	0	0
0	0	0	0	0	1
0	0	0	1	1	1
0	0	0	1	1	1
0	0	0	1	1	1
0	0	1	1	1	1

Given a picture P1 with histogram H1 and a desired histogram H2, then normalization is the transformation of P1 to the picture P2 such that P2 has histogram H2. For example, if the histogram of Table I has to be stretched to the range (0,60), the transformation

$$y = a*x + b,$$

where a=5 and b=-15, can be used. The resulting normalized

matrix is given in Table IV.

Table IV. Normalized Picture Matrix of Table I.

0	20	20	20	20	35
5	15	15	15	20	45
10	10	10	45	45	45
5	5	5	45	50	60
10	10	15	45	60	60
35	35	40	40	45	50

A Gaussian curve refers to the shape of the histogram's distribution as being Gaussian [17], e.g., see Chapter II, page 18. Similarly, a triangular curve refers to the shape of the histogram's distribution as being triangular. Flattening is defined to be the assignment of an equal number of pixels to each gray level in the full range (0,K), for a b-bit pixel. The shape of the histogram distribution is a flat line. If the total number of pixels is not divisible by the range of the gray levels, then the distribution will be uniform (within ±1). For example, for a 250*300 picture with gray scale range (0,63), the first 56 gray levels in the new picture will have 1172 pixels and the next 8 gray levels will have 1171 pixels. It should be noted that these distributions are defined for the full range. The definitions can easily be modified to suit any range (a,b), where a and b are the desired minimum and maximum gray levels. Also, flattening [20] has been defined

in terms of approximate uniform distributions, some even with missing gray levels.

The neighborhood of a pixel is defined to be the set of pixels which are inside some region containing the pixel. The region can be of any shape such as square, rectangular, diamond, etc. If a(i,j) is the pixel at the ith row and jth column, then N2 of a(i,j) is a set of four pixels [a(i,j+1), a(i,j-1), a(i-1,j), a(i+1,j)], and is sometimes referred to as the diamond neighborhood. For example in Table I, for a(2,2)=6 in the 2nd row and 2nd column, the neighborhood pixels for the diamond neighborhood are a(2,3), a(2,1), a(1,2), a(3,2) = (6,4,7,5).

A distance function d(x,y) satisfies the following relationships for any three integers a,b and c [51]:

d(a,b) = 0 if and only if a=b	positive definiteness
d(a,b) = d(b,a)	symmetry
d(a,c) ≤ d(a,b) + d(b,c)	triangular inequality

The gradient of a continuous picture is the first derivative of the picture function. For a discrete picture function [51], the gradient magnitude (grad) and direction (theta) is given by:

grad = sqrt(X**2 + Y**2),

where: X = a(i-1,j) - a(i+1,j),

Y = a(i,j+1) - a(i,j-1),

and

theta = arctan(Y/X).

A Roberts operator for a discrete picture function

[51] is defined as:

$$Rob = abs[a(i,j) - a(i+1,j+1)] + abs[a(i+1,j) - a(i,j+1)].$$

A binary gradient picture is generated by first obtaining the gradient picture of the original picture and then the gradient picture is thresholded to yield a binary picture. The picture element of this picture with gray level 1 will be called the binary edge pixel. The Roberts gradient operator was used in this thesis to obtain gradient pictures.

The Laplacian of a continuous picture is the second derivative of the picture function. For a discrete picture function [51], the Laplacian magnitude is defined as:

$$Lap = 4*a(i,j) - [a(i-1,j) + a(i+1,j) + a(i,j-1) + a(i,j+1)].$$

Derivative type operations are useful for obtaining edges and contours of pictures by first finding the gradient or Laplacian pictures, which are then thresholded. It should be noted that the gradient or the Laplacian is not computed at the edges of the picture, i.e., for an N*N picture, the output picture's size is (N-1)*(N-1).

1.3 Organization of Thesis

In Chapter II, preprocessing of pictures is discussed. Two basic concepts, namely normalization and enhancement are reviewed and the importance of each technique is justified. Two new methods have been developed for basic preprocessing steps. Results obtained on digital pictures, by using the proposed methods, as well as the existing methods are given.

Registration of pictures is considered in Chapter III. A method for automatic detection of control points is presented. An iterative feedback correction technique is developed for registration purposes. Also, a technique for registration of pictures with large scale differences is given, along with an example.

In the Chapter IV, a fast template matching method based on normalized correlation is developed. Two new algorithms for fast cross correlation are proposed. A comparison is made between the computation requirements and the results obtained by the various methods.

The problem of change detection is discussed in Chapter V. A brief survey of change detection is given, and various types of changes are defined. Different operations such as averaging, cleaning, computation of two dimensional histograms, and binary thresholding have been used to obtain difference pictures. Changes which have been obtained in five test digital pictures are given.

Finally in the last Chapter, the results are interpreted and suggestions for further improvement are given. A list of computer programs is given in Appendix A and detailed information about the pictures is summarized in Appendix B.

Chapter II

PREPROCESSING OF PICTURES

This chapter presents some preprocessing methods for computer perception of pictures. Preprocessing is a first step for obtaining important features and properties. Some of the basic techniques are normalization, enhancement, restoration, spatial filtering, segmentation, and contrast stretching; however, only normalization and enhancement techniques will be considered here.

The first preprocessing method deals with gray scale normalization. It transforms the gray level histogram of a picture to any desired distribution. The transformation retains the spatial relationship of objects in the pictures. The second method, picture enhancement, is based on binary neighbor relations among the gray levels of the pixels. Applications of this approach include improved edge and contour detection, and the generation of uniform regions. In the last section, the techniques developed are applied to several pictures and a detailed comparison is made with other preprocessing methods.

2.1 Gray level Normalization

If two data vectors are obtained by different

techniques and if they have to be compared (e.g., their similarity), the data vectors are usually normalized. This type of problem frequently occurs in statistics, pattern recognition and social science applications. Some common ways to achieve this is to either subtract the mean of the vectors and divide by the standard deviation or just divide each data component by the standard deviation. Normalization is essential for pictures taken at different times and under different conditions, since the gray levels will be quite different, even if the overlapping sub-scenes are essentially the same. For example, to detect differences between two pictures, it is important either to normalize both pictures to some standard picture or to normalize one picture relative to the other [25]. Normalization can be done by linear stretching of different parts of the histogram range. This can be achieved by a linear operation like

$$y = a*x + b,$$

where a,b are normalizing factors and x,y are gray levels. This approach may not be sufficient for all pictures, because the desired histogram shape may not be always obtained by linear operations.

Kruger [34] suggests a linear operation for histogram normalization. The cumulative frequency distribution of the picture is computed. If hist(i) represents the frequency for gray level i, a simple operation like

hist(i) <-- hist(i) + hist(i-1), for i=2,3,...,63,

gives the cumulative distribution. The cumulative distribution is then scaled to give a transformation vector, such that hist(63) is 63 (for a 6-bit pixel). The normalized picture is obtained by applying the transformation vector to the picture. The method is simple to use, but has the problem that, if the histogram distribution starts from gray level 32 (on a 0 to 63 scale), the normalized picture will be too white and the features will not be properly identified.

Haralick et al [23] present an algorithm, in which gray levels are grouped such that the total number of gray levels is reduced to N levels. The algorithm first computes the cumulative frequency distribution, which is then normalized to the range (0,1). The grouping of levels is obtained by dividing the total cumulative distribution into N levels. This yields a transformation vector, which when applied to the gray levels of the picture, gives a normalized picture. This kind of requantization of gray levels is good for pictures which are to serve as input for some feature extraction algorithms. The feature extraction algorithms expect the picture to be noise free and to have a smaller number of gray levels for reducing the dimensions of the feature vector space.

2.1.1 Arbitrary Histogram Shape Method [32]

Given a picture P1 with histogram H1, and a desired histogram H2, the proposed method transforms P1 to a picture

P2 such that P2 has histogram H2. The transformation preserves spatial relationships among the objects in the picture. In other words, the transformed picture looks similar to the original, except that features may get enhanced. The method is non-iterative, and expands the original gray level range to the full range (0,K), for a b-bit pixel. Unlike other techniques, in which the number of gray levels is either decreased or left the same, this method expands the original range to the full range (0,K). For example, flattening differs from that of Eberlein et al [15] in the sense that the full range (0,K) gets approximately the same number of pixels. It should be noted that the method generates artificial gray levels in the transformation, i.e., new gray levels are introduced in the picture.

Definition 1: A decomposition matrix D(k,i) is a 2 dimensional matrix in which the row and column pointers k and i correspond to the desired and original gray level range. The first row corresponds to the gray level frequency vector of the original picture and the first column corresponds to the desired gray level frequency vector. The next 64 columns (for a 6-bit pixel) correspond to the decomposition of the original frequency into the desired groups. The vertical sum of the columns equals the frequency of gray levels of the original picture and the horizontal sum equals the frequency of gray levels of the

desired picture.

To illustrate the definition, consider the normalization of the Neuron picture given in Fig. 1, such that the histogram will have a Gaussian shape, as in the transformed picture given in Fig. 2. The histogram H1 of the original Neuron picture is given by Fig. 3. The frequency distributions of H1 and the desired curves H2 are given by:

Original histogram = (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
1,5,46,814,1757,2511,3049,3441,4247,5412,6297,
6591,6846,7006,5498) for the first 32 gray levels.

Desired histogram = (20,27,37,49,65,85,110,141,178,223,277,
340,414,498,593,700,817,945,1081,1225,1375,1527,1679,
1828,1970,2103,2222,2324,2407,2468,2505,2535).

Because of the symmetry in a Gaussian distribution, only the first 32 gray levels are given. The desired histogram vector H2, see Fig. 4, was obtained from the Gaussian distribution [17]. The decomposition matrix for gray levels 17 to 20 is given in Table V. From Table V, the column for gray level 19 reads: decompose the 46 pixels of gray level 19 of P1 into 14 pixels of gray level 0, 27 pixels of gray level 1 and 5 pixels of gray level 2. The Neuron picture with a flat distribution is given in Fig. 5. The proposed method is described in algorithm A.

Algorithm A:

Step 1: Compute the histogram of the original picture.

Step 2: Compute the desired histogram of the new picture.

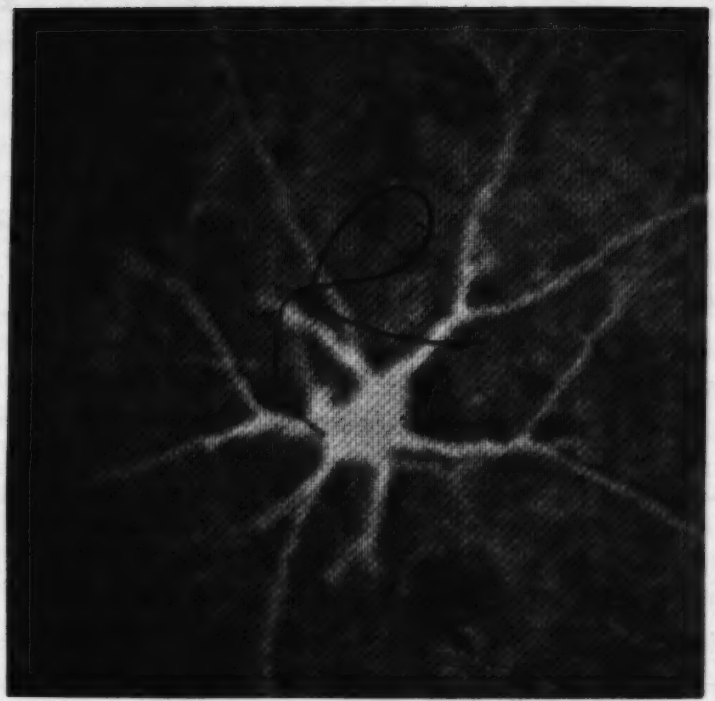

Fig. 1. Neuron.

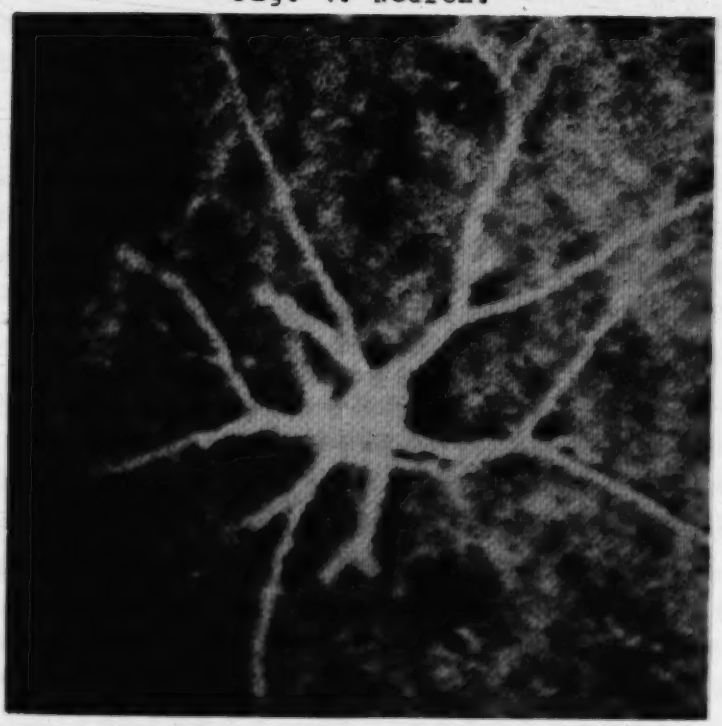

Fig. 2. Neuron with Gaussian histogram.

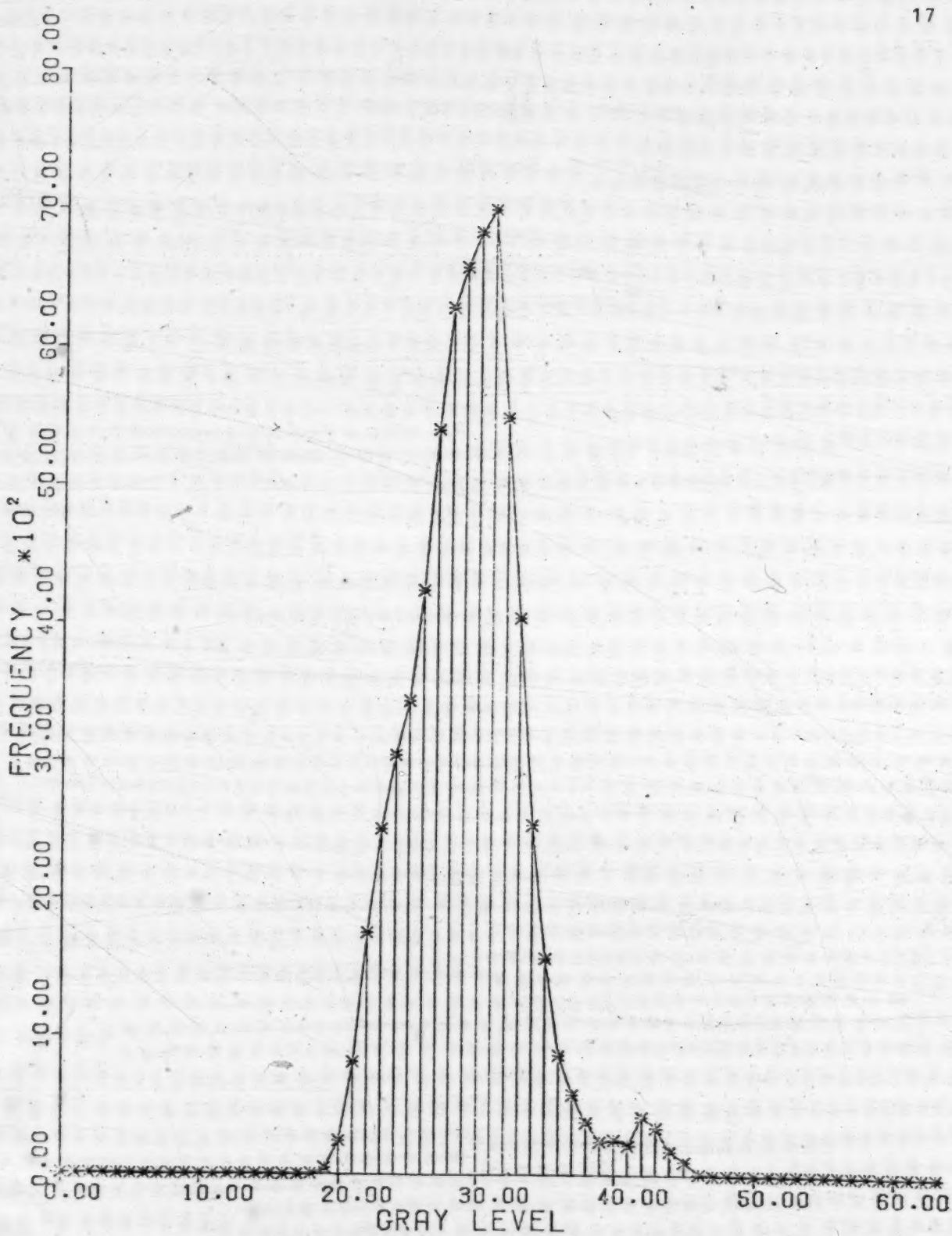

Fig. 3. Histogram of Neuron.

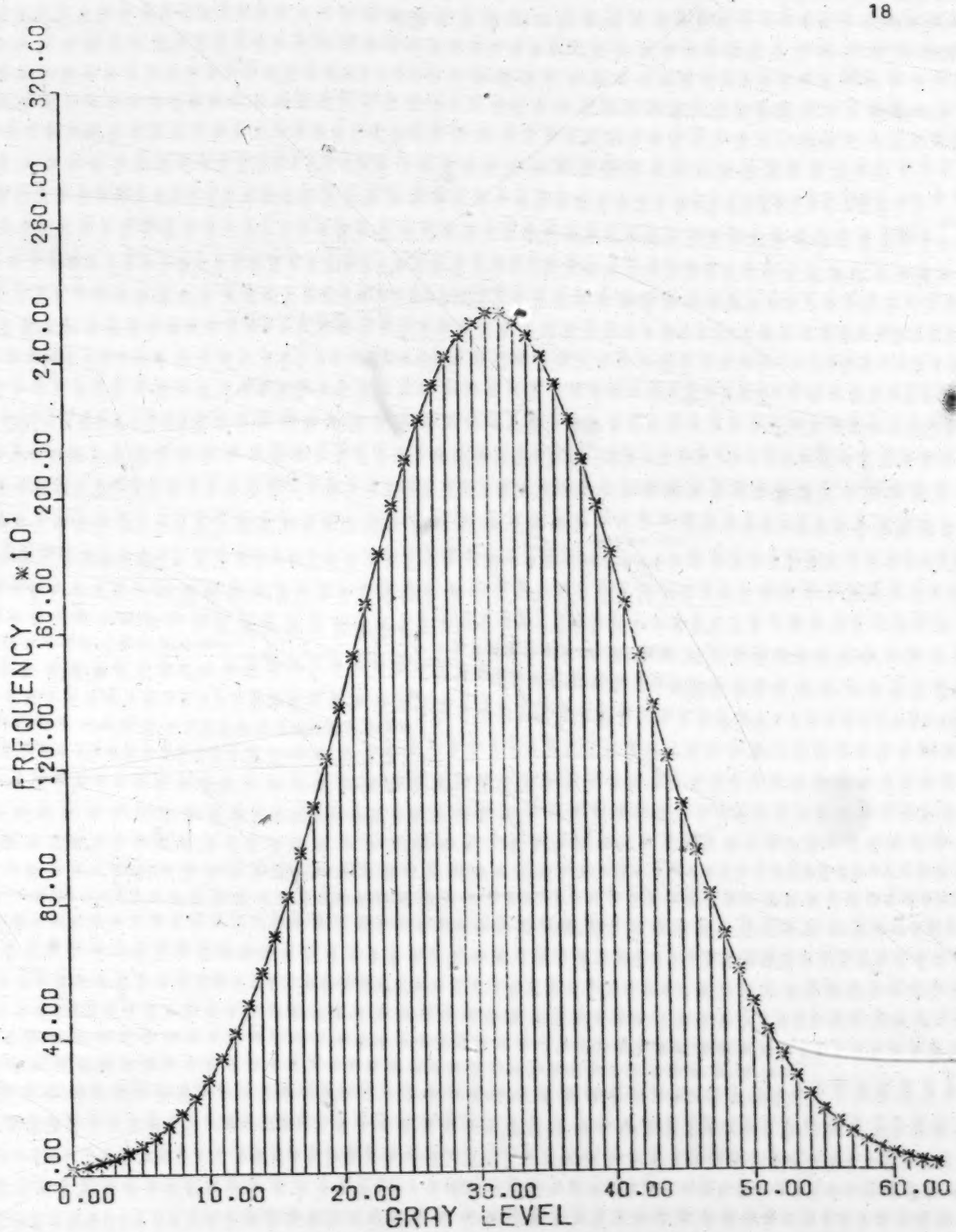

Fig. 4. Normalized Gaussian histogram.

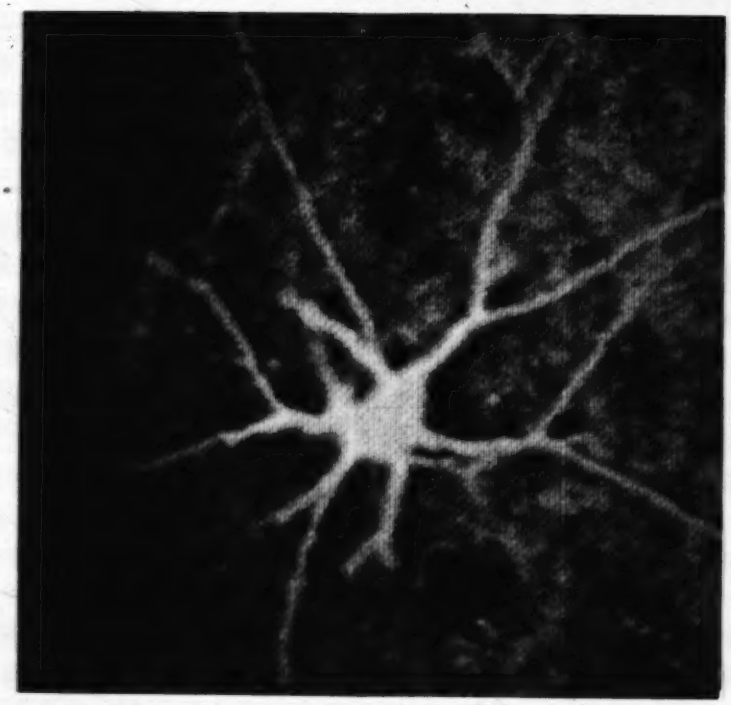

FIG. 5. Neuron with flat histogram.

Table V. Decomposition Matrix.

	-1	0	1	2	...	17	18	19	20	...	63
-1	0	0	0	0	...	1	5	46	814	...	
0	20	0	0	0	...	1	5	14	0	...	
1	27	0	0	0	...	0	0	27	0	...	
2	37	0	0	0	...	0	0	5	32	...	
3	49	0	0	0	...	0	0	0	49	...	
4	65	0	0	0	...	0	0	0	65	...	
5	85	0	0	0	...	0	0	0	85	...	
6	110	0	0	0	...	0	0	0	110	...	
7	141	0	0	0	...	0	0	0	141	...	
8	178	0	0	0	...	0	0	0	178	...	
9	223	0	0	0	...	0	0	0	154	...	
...				.	.	.					
63				.	.	.					

Step 3: Compute the decomposition matrix, i.e., break the frequency of every gray level present in the original picture into a group. The number and the assignment of pixels to gray levels depends on the desired histogram distribution. For example, the first non-zero gray level is 17 and the number of pixels for gray level 0 should be 20 in the desired histogram. Thus one pixel of gray level 17, 5 pixels of gray level 18 and 14 pixels of gray level 19 will be used to generate the 20 pixels with gray level 0 in the new picture. Note that the remaining 32 pixels of gray level 19 will also be used for further decomposition.

Step 4: Starting from row number 1, repeat the following step row by row, for all rows. Compute the average gray level of the current pixel. The average is obtained by summing gray levels in the diamond

neighborhcod of the current pixel including the gray level of current pixel. The sum is then divided by 5. If this average is greater than the current pixel's gray level, then assign the highest possible gray level from the row pointer of the decomposition matrix, otherwise assign the lowest possible gray level. If any desired gray level frequency is zero, choose the next one. Subtract one from the assigned gray level in the corresponding matrix element.

In step 2, the desired histogram's distribution can be modified so that the range is not (0,K), but may be fixed at (a,b), where a,b are the minimum and maximum gray levels desired in the picture. Also, once step 4 is done, all the entries (except the first row and column) in the decomposition matrix will be zero.

It is important to obtain more <u>contexual</u> information for proper assignment of gray levels. The simple technique of averaging worked well for the test pictures, but in other applications, the algorithm may have to be modified to include this information. For example, the problem of reconstructing a gray level picture from a binary picture was attempted and the results were not impressive. It was assumed that only a binary picture is available and other information like histogram distribution, mean, variance etc., was not known. If reconstruction can be done with sufficient fidelity for some pictures, then data compression of b:1 for a b-bit pixel can be achieved.

2.2 Picture Enhancement

Picture enhancement is concerned with making distinguishable details more visible in a picture. Two different approaches to enhancement exist, namely smoothing and sharpening. Picture smoothing is a blurring operation, usually obtained by some method of averaging. Smoothing evens out sharp transitions in gray levels produced by noise. Picture sharpening, on the other hand, is a deblurring operation, where the rate of change in the gray levels should be large. Noisy pictures should not be sharpened, since the noise itself will get amplified. Ideally, pictures should be sharpened without losing information; however, most enhancement methods do lose some information. In this section, only picture sharpening will be studied.

One of the common techniques for sharpening is the use of derivative type operations like the gradient or the Laplacian [51]. Weszka et al [63] simply add a Laplacian to the picture for enhancement. The operator is given by:

$$2*a(i,j) - Lap,$$

where $a(i,j)$ is the current pixel and Lap is the magnitude of the Laplacian at pixel $a(i,j)$, see Section 1.2. The derivative type of operation can be implemented either digitally or optically. Singular value decomposition [2] has recently been used for enhancement, in addition to its applications in ccding and restoration. The picture is decomposed into a set of eigenimages of rank 1. A linear and non-linear weighted filter is then applied to this

decomposition, which enhances certain eigenimages, also called outer products. The enhancement predominantly sharpens edges in pictures.

Also, a non-linear transformation for enhancing digital pictures has been proposed by Kramer and Bruckner [33]. Given a digital picture P, a new picture is obtained by replacing the gray levels of each pixel by either a local maximum or minimum within a neighborhood, whichever is closer to the gray level of the current pixel. The process is repeated until successive pictures are approximately the same, implying convergence. This process is iterative and according to the authors, it takes about 20 to 50 iterations for a 27*33 matrix. For practical applications like LANDSAT pictures, where one scene consists of 2286*3600*4 pixels, iterating many times is extremely time consuming. It follows, therefore, that it would be more useful to be able to enhance in a single transformation, and hence require only one or two passes of the picture. The purpose of this section is to propose a non-iterative procedure for picture enhancement.

2.2.1 The Proposed Method [32]

Neighborhood information has been used very often in picture processing [23]. For example, the notion of co-occurences of gray levels and various other forms of statistics have been exploited for texture analysis. The neighborhood matrix of [23] differs from the matrix

introduced in this section in the sense that no statistics are computed. The binary matrix obtained from gray levels is defined formally in the following:

Definition 2: Let y be the gray level at the point x, N(x) represent the set of gray levels occurring in the neighborhood of x, and z be any gray level of the picture set, then a binary neighborhood matrix is defined as follows:

1. If $z \in N(x)$ then NEIG(y,z) = 1,
   otherwise NEIG(y,z) = 0.
2. If $z \in N(x)$ then NEIG(z,y) = 1,
   otherwise NEIG(z,y) = 0.
3. NEIG(z,z) = 1.

The enhancement method is described in the following algorithm.

Algorithm B:

Step 1: Compute the binary neighborhood matrix. Next, each row of the matrix is scanned for the leftmost and rightmost '1'. The object is to find the minimum and maximum gray levels, which are neighbors to the current gray level(or row). The gray level (minimum or maximum) which is closest to the row number is assigned to the transformed vector.

Step 2: The range of the original gray levels is divided into two parts (low range and high range) at the middle. If any transformed gray level within the low range is greater than the mid-range value, then

that transformed gray level is replaced by the minimum (or leftmost) neighbor of the current row. Similarly, for the high range, if any transformed gray level is less than the mid-range, the transformed gray level is replaced by the maximum (or rightmost) neighbor.

Step 3: Since the transformation is many to one, a condition where the gray levels with large frequencies are mapped into one single gray level may occur. In that case, a check is made to determine the number of pixels which the transformed picture contains. If this number is more than P% of the total number of pixels, these gray levels are not transformed.

Step 4: The enhanced picture is obtained by applying the transformation vector so obtained to the original picture.

The binary neighborhood matrix is symmetric, thus only half of the matrix has to be computed. To illustrate the definition, consider the matrix of gray levels in Table VI.

Table VI. Original Matrix.

18	21	24	24	21
15	12	25	21	21
16	16	25	25	25
16	17	18	25	21

Let the pixel under consideration be a(2,2), with the neighborhood function N2, where

$$N2(x,y) = [\,(x,y),(x-1,y),(x,y-1),(x+1,y),(x,y+1)\,].$$

The following relations are defined for gray level 12 in the binary matrix:

$NEIG(12,z) = 1$, $NEIG(z,12) = 1$, $NEIG(12,12) = 1$, where $z = (21,15,16,25)$.

Similarly after scanning rows 2 to 3, and columns 2 to 4, the binary matrix of Table VII is obtained.

Table VII. Binary Neighborhood Matrix.

Gray level	12	15	16	17	18	21	24	25	Transformed Vector
12	1	1	1	0	0	1	0	1	12
15	1	1	0	0	0	0	0	0	12
16	1	0	1	1	0	0	0	1	12
17	0	0	1	1	0	0	0	0	17
18	0	0	0	0	1	0	0	1	18
21	1	0	0	0	0	1	1	1	25
24	0	0	0	0	0	1	1	1	25
25	1	0	1	0	1	1	1	1	25

The enhanced matrix of Table VI is given by Table VIII.

Table VIII. Enhanced Matrix.

18	25	25	25	25
12	12	25	25	25
12	12	25	25	25
12	17	18	25	25

The new method is compared with Kramer and Bruckner's method and Weszka et al's method [63] in the next section.

2.3 Conclusions and Test Results

For comparing the results produced by different methods, the Neuron and Spinach pictures were used as test pictures, see Appendix B for picture details. The pictures

are given in Fig's 1 and 6 and the histograms are given in Fig's 3 and 7 respectively. The following definitions are introduced for connectivity and the degree of connectedness.

Definition 3: A binary edge pixel $p(i,j)$ is said to be connected to another binary edge pixel $q(k,m)$, if the following conditions are satisfied:

1. $p(i,j) = q(k,m) = 1$.
2. $k = i-1$, i, or $i+1$.
3. $m = j-1$, j, or $j+1$.
4. If $k = i$ then $m \neq j$ and vice versa.

Definition 4: The degree of connectedness in a picture is the percentage of the number of connected edge elements with respect to the total number of edge elements.

The following criterion will be used to compare the Spinach and Neuron pictures obtained by different methods.

1. Visual inspection of original pictures.
2. Visual inspection of edge pictures.
3. Total number of edge elements.
4. Total number of connected edge elements.
5. The degree of connectedness.

The edge pictures were obtained by using a Roberts operator. The number of edge elements and the connected edge elements are given in Table IX and Table X for Spinach and Neuron respectively.

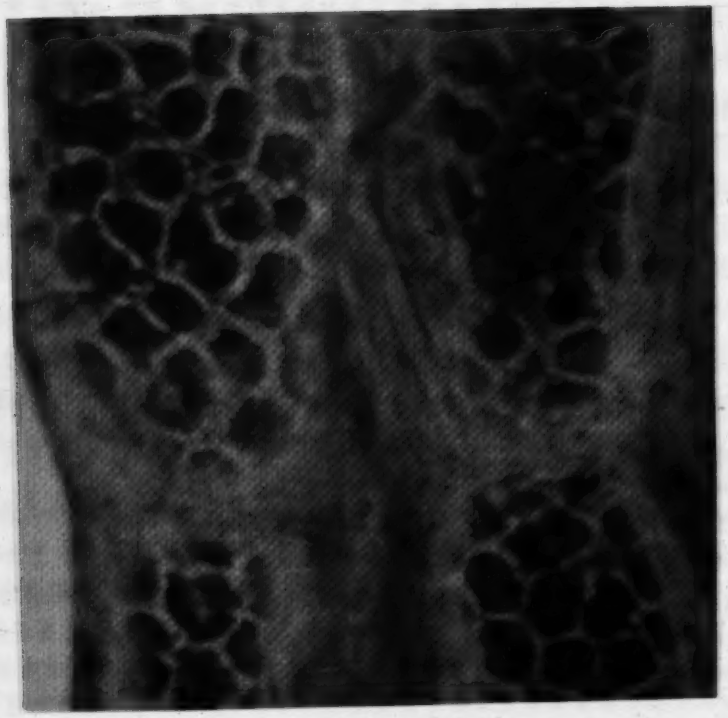

Fig. 6. Spinach.

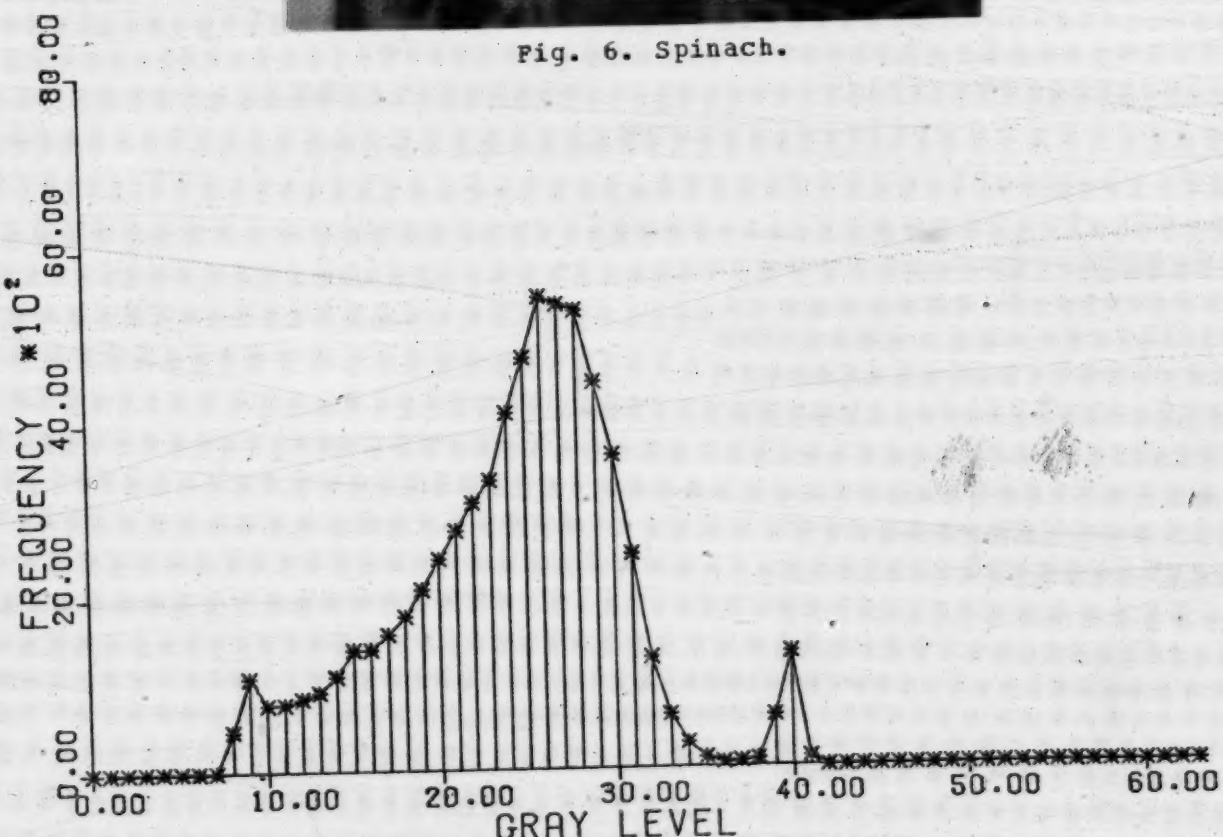

Fig. 7. Histogram of Spinach.

Table IX. Degree of Connectedness for Spinach.

Name	Edges	Connected Edges	Degree	Threshold for Roberts operator
Original	4455	3539	79.43	7
Enhanced	9647	8635	89.50	7
Flat	4958	3525	71.09	25
	8471	6428	75.88	20
	14776	12182	82.40	15
Gaussian	791	447	56.50	20
	8394	6306	75.12	10
	17831	14547	81.50	7
Weszka	10845	8323	76.74	7

Table X. Degree of Connectedness for Neuron.

Name	Edges	Connected Edges	Degree	Threshold for Roberts operator
Original	630	424	67.30	7
Enhanced	10483	9454	90.18	7
Flat	4466	2390	53.51	25
	9078	5906	65.05	20
	17187	13588	79.05	15
	29757	27062	90.94	10
Gaussian	389	176	45.24	20
	9184	5566	60.60	10
	21795	17128	78.58	7
Weszka	4763	2445	51.33	7

2.3.1 Gray Scale Normalization

The pictures with flat and Gaussian histograms are given in Fig's 8 and 9 for Spinach and Fig's 5 and 2 for Neuron respectively. Normalization enhances the pictures, while some minute detail is lost. By re-distributing the gray levels among the pixels, contrast is improved, thus objects which could not be seen earlier, are evident now. The edge pictures of the original and normalized pictures, flat and Gaussian, are given in Fig's 10-12 for Neuron and in Fig's 13-15 for Spinach. Visually, the edges of the normalized pictures look better than the originals. But the normalized edge pictures contain too much noise. This is also evident empirically from the degree of connectedness, see Tables IX and X. For obtaining edges of the normalized pictures, the threshold in the Roberts operator was varied from 7 to 25, giving different degrees of connectedness. It should be noted that the concept, that the higher the degree the better the edge picture would be, is not true. In fact for a threshold 7, the normalized edge pictures were thick and noisy. Although it is desirable to have a higher degree, the visual output should also be taken into consideration.

Kruger's method [34] was applied to both test pictures. The processed pictures look approximately the same as the pictures with flat histograms. The pictures were also normalized with a triangular shaped histogram. The resulting pictures look similar to pictures with

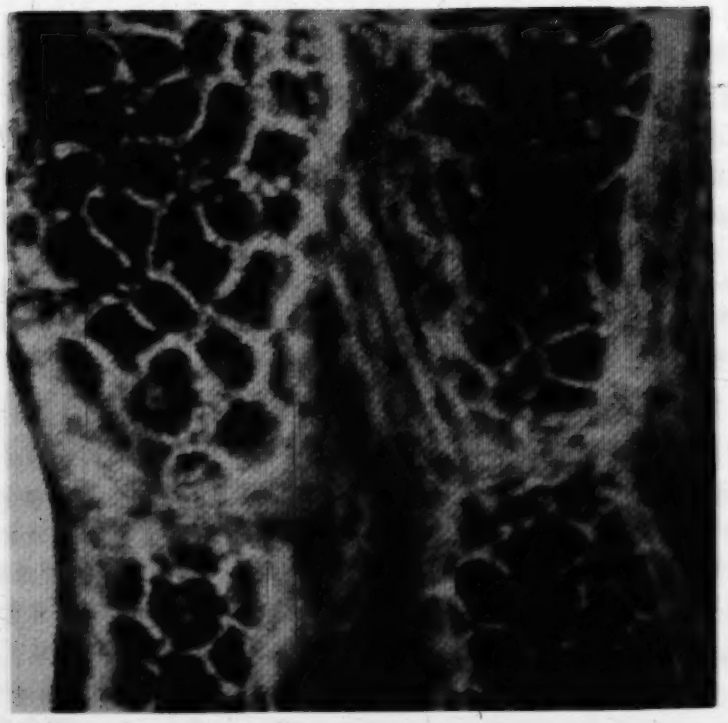

Fig. 8. Spinach with flat histogram.

Fig. 9. Spinach with Gaussian histogram.

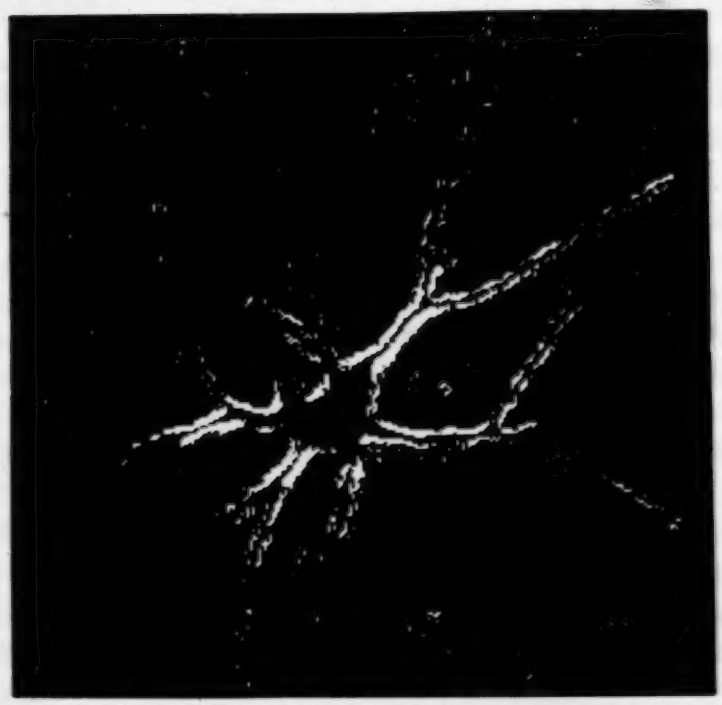

Fig. 10. Edges of Neuron.

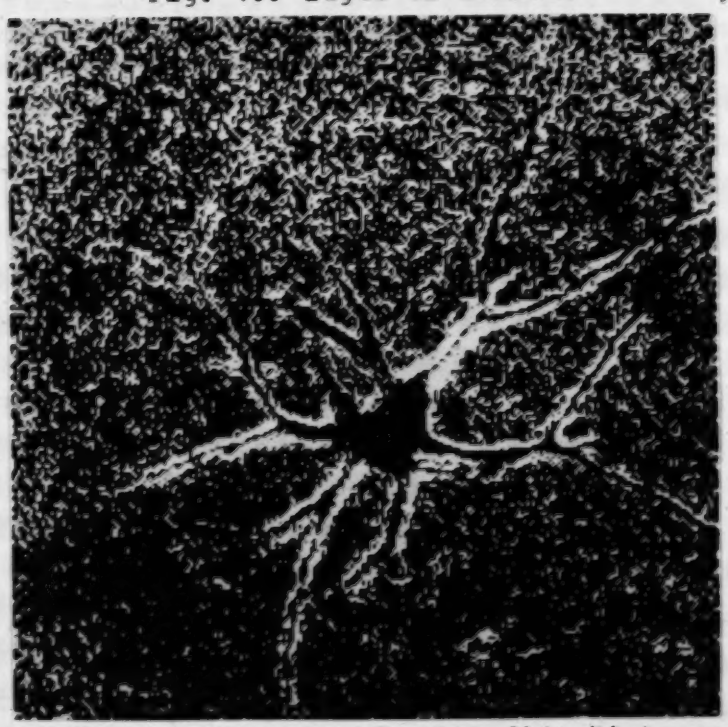

Fig. 11. Edges of Neuron with flat histogram.

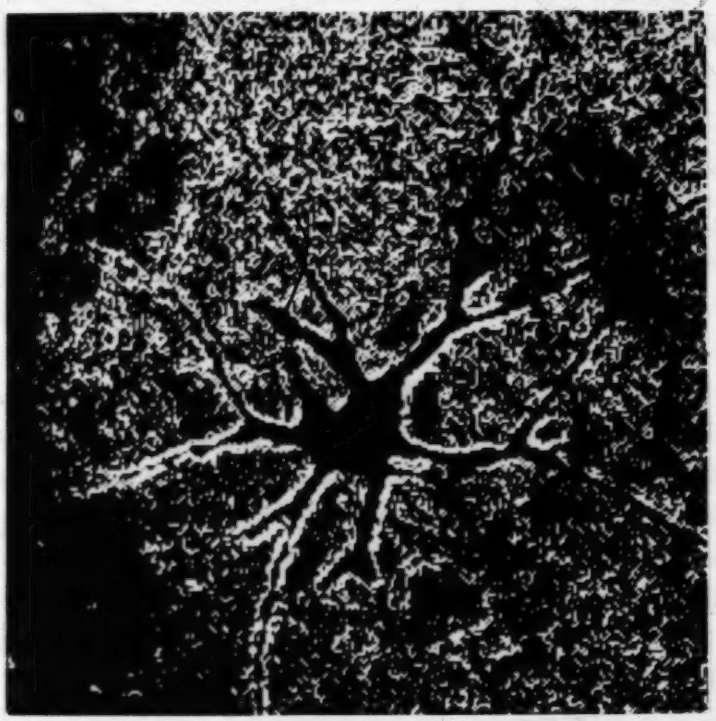

Fig. 12. Edges of Neuron with Gaussian histogram.

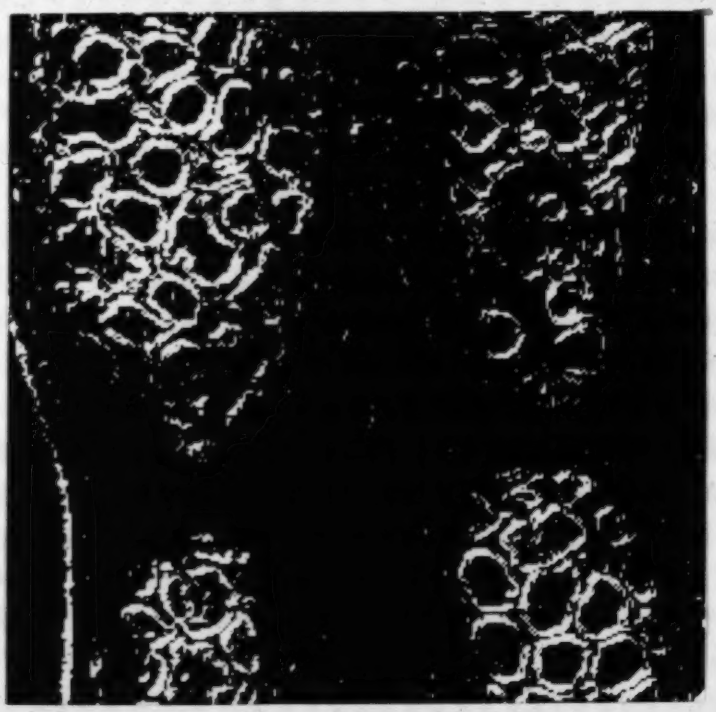

Fig. 13. Edges of Spinach.

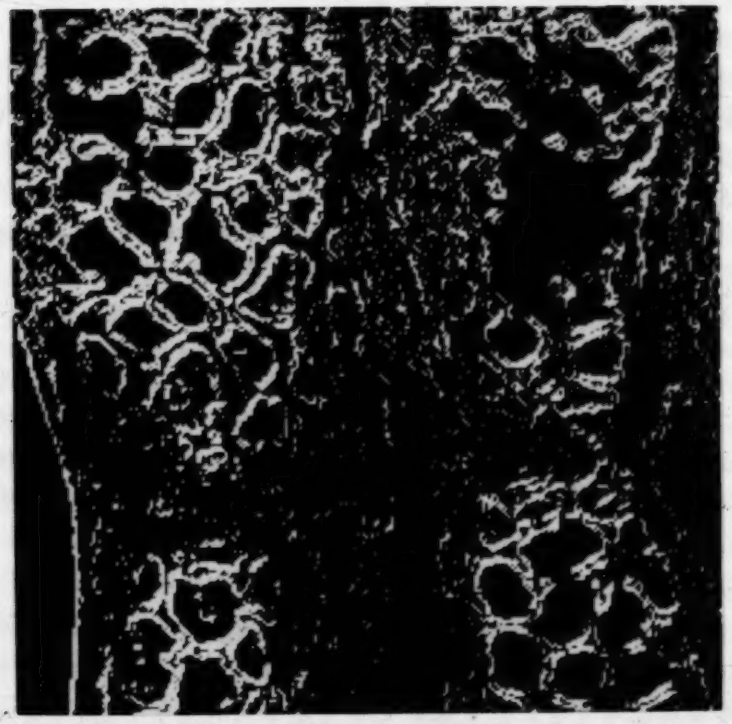

Fig. 14. Edges of Spinach with flat histogram.

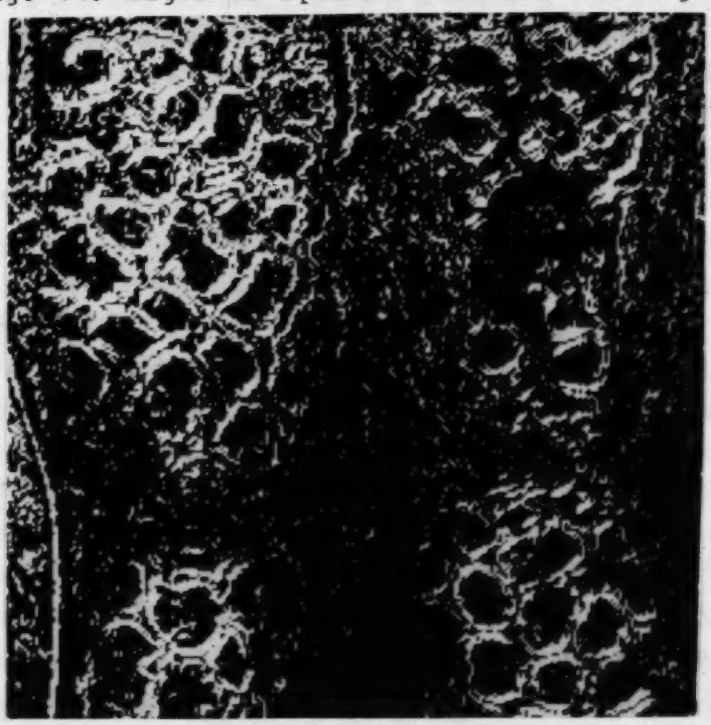

Fig. 15. Edges of Spinach with Gaussian histogram.

Gaussian histograms. On comparing the difference between two normalized pictures (flat and Gaussian), it was noted that visually, a flat distribution enhances much more than a Gaussian distribution and empirically the edge pictures of flat distributions have less noise than the edge pictures of Gaussian distributions. Since the motivation for designing the normalization technique was to generate standard pictures, which were obtained during different conditions, the Gaussian distribution was chosen for the standard. The Gaussian distribution was chosen because 1) it enhances less than the flat and 2) the majority of the pixel's values are near the mid-range. Further examples of Gaussian normalization can also be found in other pictures, given in pages 117, 123, 125, 126, 128 and 129. It should be noted that in these cases, the range of normalization was (0,255).

2.3.2 Picture Enhancement

Kramer and Bruckner's iterative method was programmed for a maximum of six iterations. It was found that the pictures got worse instead of enhancing them as shown in Fig's 16 and 17 and the shapes of the objects in both pictures were smeared. Because of this smearing, the edge pictures and the degrees of connectedness were not computed.

Weszka et al's method also did not sharpen pictures as shown in Fig's 18 and 19. The edge pictures are given in Fig's 20 and 21 for Neuron and Spinach respectively.

Fig. 16. Kramer Neuron.

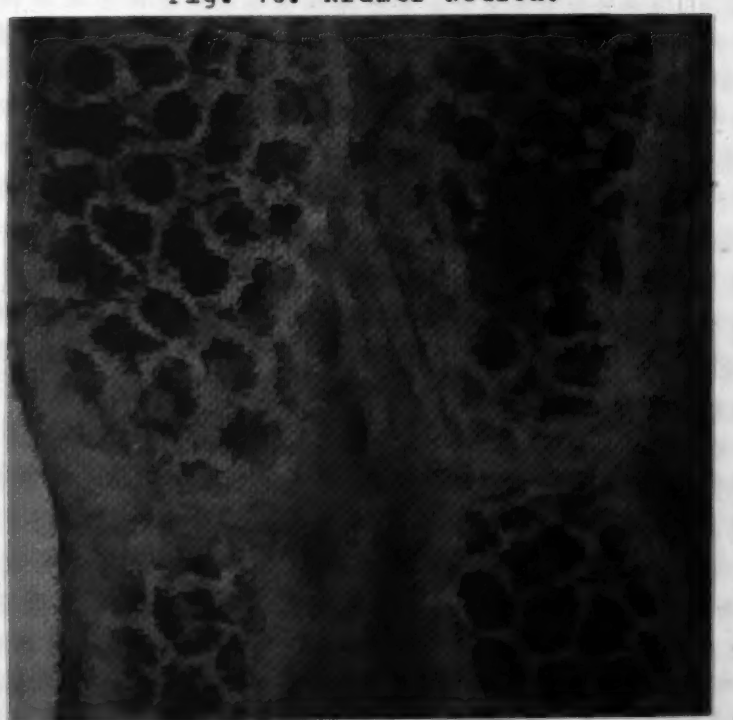

Fig. 17. Kramer Spinach.

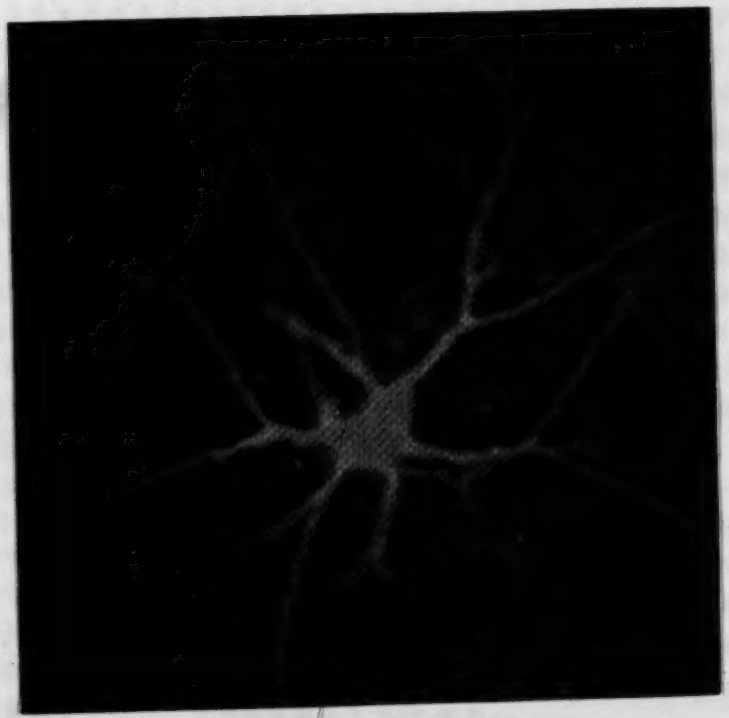

Fig. 18. Weszka Neuron.

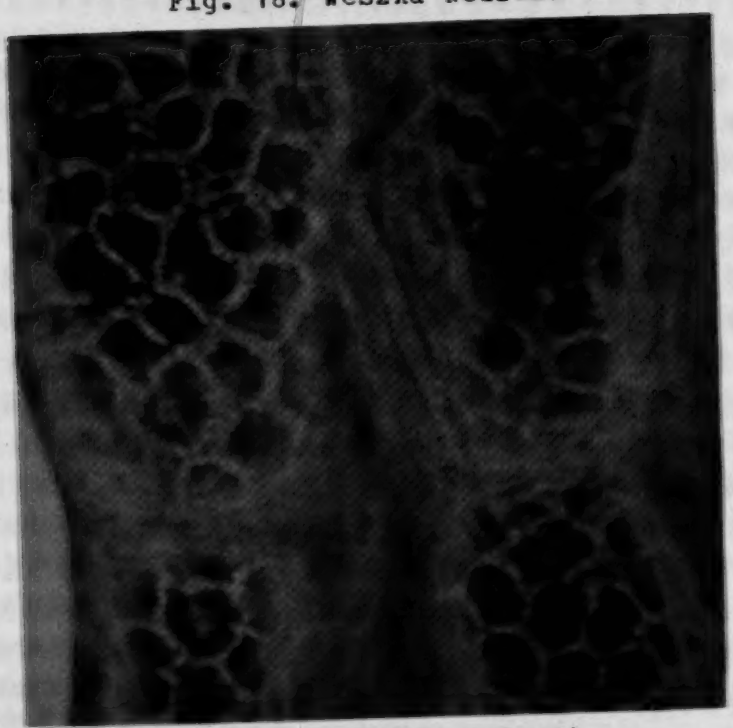

Fig. 19. Weszka Spinach.

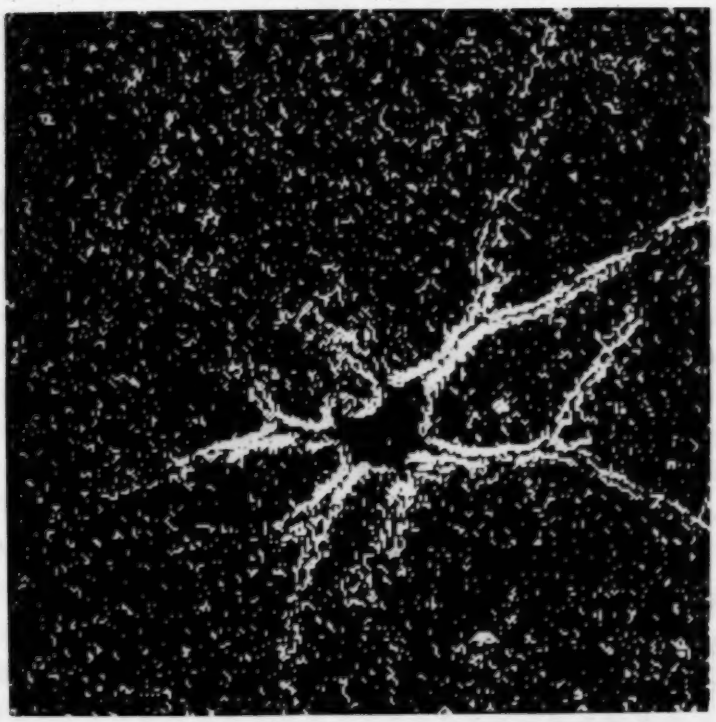

Fig. 20. Edges of Weszka Neuron.

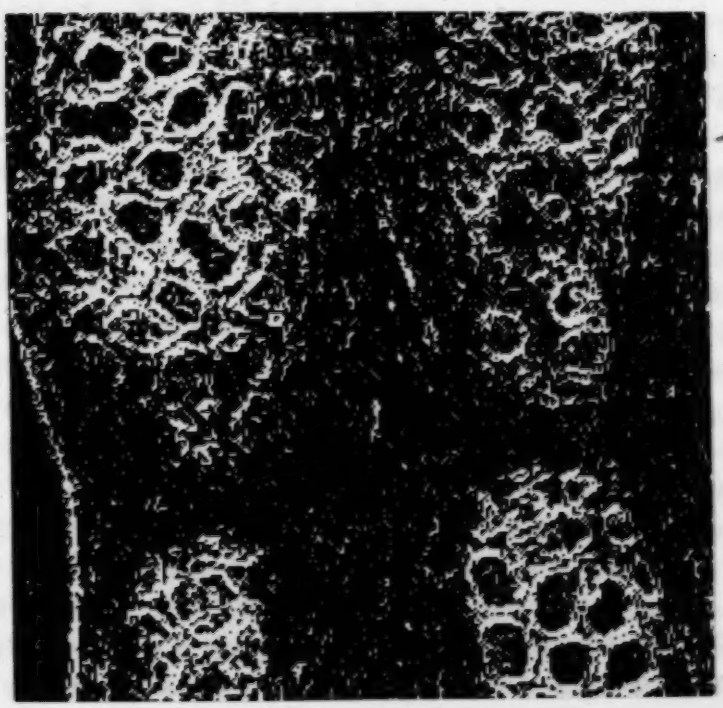

Fig. 21. Edges of Weszka Spinach.

Visually, the edge boundaries look fuzzy and the shape is either broken or thick. Also the edge pictures are noisy. Empirically the degree of connectedness is low for both test pictures. The surprising thing is that the degree of connectedness is higher in the original edge pictures than in the enhanced pictures.

The enhanced pictures obtained by the new method are given in Fig's 22 and 23. Visually, in the enhanced pictures, small objects which are hard to see in the original pictures, are clearly evident. As the method reduces the number of gray levels, the enhanced pictures appear to be bilevel. This feature may not be desirable in certain applications. The edges of the enhanced pictures are given in Fig's 24 and 25 for Neuron and Spinach respectively. The edges look very sharp, noise free and much better than those detected on either the original or normalized pictures. The objects in the edge pictures have clearly delineated boundaries. The degree of connectedness of both edge pictures is the highest, which also shows that noise problem is not severe. Thus for noisy pictures, this enhancement method is the best. The analysis was repeated with other standard edge operators such as the gradient and the Laplacian. The results were consistent in all cases.

Fig. 22. Enhanced Neuron.

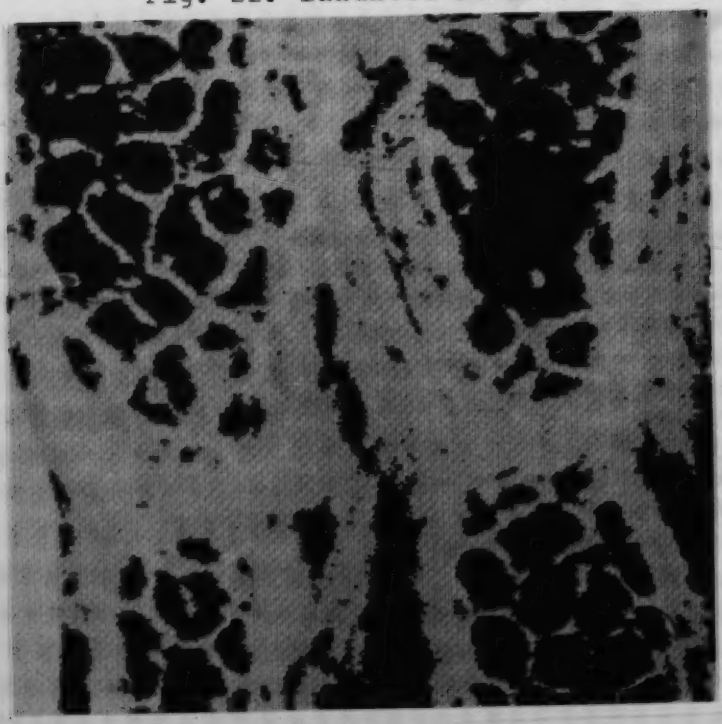

Fig. 23. Enhanced Spinach.

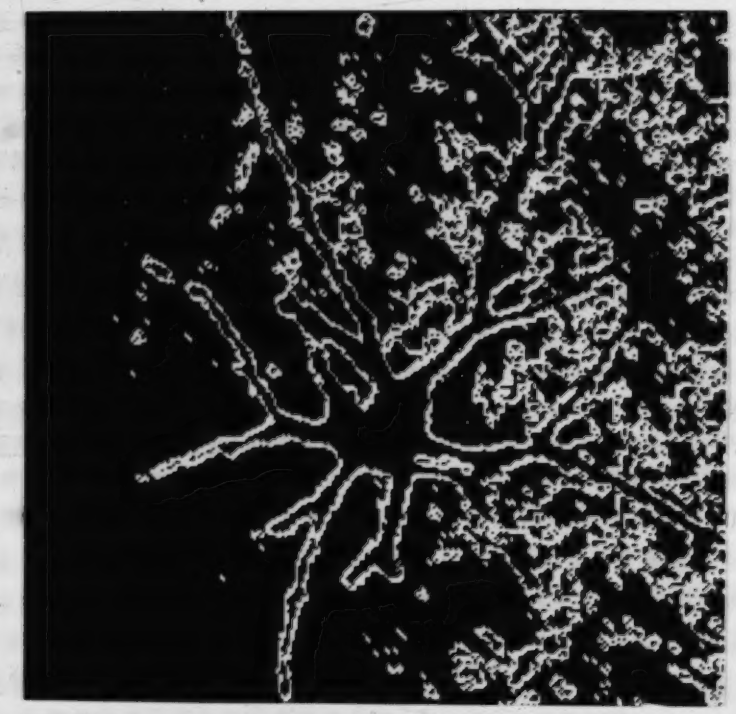

Fig. 24. Edges of enhanced Neuron.

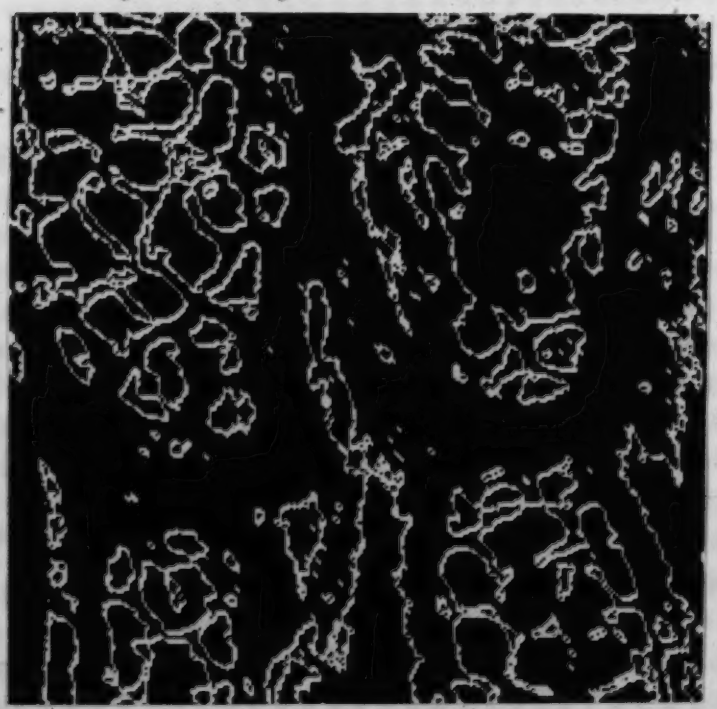

Fig. 25. Edges of enhanced Spinach.

Chapter III

REGISTRATION

This chapter deals with the proper alignment of two pictures, when both pictures may have been obtained from a different angle, height, or orientation. For example, the aircraft or satellite's position and orientation will vary with different scans of the same scene. The two pictures must then be properly aligned by some transformation such as affine, projective, etc. The alignment is also necessary if changes between two successive pictures of the same scene are to be obtained. For measuring the accuracy of registration, the two pictures, in whole or in part, can be compared by some similarity measure. It is also possible to extract a window or sub-picture from the first picture and try to locate the corresponding window in the second picture. The difference between the coordinates of the original window and where it matches in the second picture, should give the registration error. It should also be noted that the registration accuracy will vary from pixel to pixel.

Once the transformation between two pictures is known, resampling techniques such as nearest neighbor, bilinear interpolation and cubic convolution [7,46,47,54,59] can be

used for generating a new version of one of the pictures, such that this new version is properly aligned with the second. The nearest neighbor technique simply rounds the row and column coordinates, i.e., selects the closest pixel. In bilinear interpolation, a two dimensional interpolation function is used to compute the gray level of the output pixel. In cubic convolution, a two dimensional resampling filter, which uses 16 input pixels, is applied to compute the output gray level. This latter technique is the best resampling technique among the cubic interpolation methods for picture transformation; however, it is also very expensive.

3.1 Introduction

In picture matching, a window or template is taken from the original picture and is positioned at the top left position of the second picture. The window and the corresponding sub-picture are then compared by some similarity measure. The window is then shifted by a displacement to the right and the similarity measure is again obtained. This process is repeated for all possible displacements in both row and column directions. The largest similarity measure of all the values obtained gives the best match point for that specific window.

The choice of the similarity measure is important. The first is the absolute difference given by:

$$R = \text{Sum } \text{abs}(x - y),$$

where x,y are the two data vectors. This measure can be normalized either by subtracting the mean or dividing by the total sample size. If gray levels of the corresponding pixels of two pictures are related by a linear relation

$$y = a*x + b,$$

then normalization is essential for taking into account factors such as gain a and offset b. The absolute difference does not take constant offset intc consideration. Therefore, difference measures should be used carefully.

The second important measure is correlation. The normalized correlation between two data vectors x and y is defined as:

$$R = \frac{\text{Sum } \{(x - \text{xmean})*(y - \text{ymean})\}}{\text{Sqrt } \{((\text{Sum}(x - \text{xmean})**2)*(\text{Sum}(y - \text{ymean})**2)\}},$$

where xmean and ymean are the means of the x and y samples respectively, see [22,53] for variations.

One of the main reasons for the popularity of the above normalized correlation is that both gain and offset factors are taken care of by subtraction of means and division by variances. This often leads to the choice of a normalized correlation coefficient as a similarity measure for matching two pictures.

In the next section, current registration techniques are surveyed. In section 3.3, a technique for the automatic selection of control points is developed. An iterative hybrid approach using both control points and correlation to produce a feedback correction is given in section 3.4. A

technique for the registration of pictures with large scale differences is given in section 3.5. In the last section, a comparison of the methods presented in the chapter is made and the results obtained are discussed in detail.

3.2 Survey of Registration Methods

Registration of pictures can be done via the control point approach or the correlation approach. In the control point technique [7,13,42,43,64], using identifiable features, a number of control points from both pictures are selected. The row and column numbers for each control point are obtained from either line printer output or direct photographs. A least squares analysis of the corresponding control point pairs gives the coefficients of the transformation function F. If vectors u,v are the row and column coordinates of the control points of the first picture and vectors x,y are the coordinates of the corresponding control points of the second picture, then the two pictures can be related by a first order transformation F(a,b,c,d,e,f), which is given by:

$$u = a*x + b*y + c.$$

$$v = d*x + e*y + f.$$

The second order transformation is given by:

$$u = a*x^2 + b*y^2 + c*x*y + d*x + e*y + f.$$

$$v = g*x^2 + h*y^2 + i*x*y + j*x + k*y + m.$$

If parallel lines remain parallel in the second picture, then a first order transformation is sufficient,

see [43]. On the other hand, for the non-parallel case and other non-linear distortions, a higher order transformation must be used.

The registered picture is then generated from the unregistered picture by applying the obtained transformation F. The main problem with this approach is that the accuracy of the transformation is directly dependent on the accuracy of the control points. In experiments with test pictures, performed by the author, the row and column coordinates of the control points were modified to introduce errors of up to 10 pixels. These incorrect control points were then used for registration of test pictures, producing a registration error of more than 5 pixels.

Kaneko [28] presents an excellent example of registration by the correlation approach, see [4,5,37] for variations. The two pictures are decomposed into 60 sets of overlapping sub-pictures. A sub-picture from the reference picture is cross correlated with the second picture and local shift parameters are obtained by a modified sequential similarity detection algorithm, see Chapter IV or [6]. The shift parameters are obtained for all the sub-pictures. A least squares analysis is then used to compute the best fit linear transformation between the pictures. The registration accuracy is reported to be about 1 pixel. It should be noted that since computing the two dimensional correlation function is very expensive, the calculations can be speeded up by a fast fourier transform, see [4,37].

Jayroe et al [27] have proposed a registration procedure based on binary gradient pictures. The binary pictures are obtained by thresholding the gradient pictures with respect to some threshold and then the binary pictures are registered by translation, rotation, and scaling of sub-pictures. Binary correlation is much cheaper than gray level correlation; however, the cost of obtaining binary pictures may offset the desired saving.

For translational alignment in cloud tracking, Smith and Phillips [56] and Leese et al [37] have used a two dimensional correlation technique. A two dimensional cross correlation matrix is computed for every possible displacement between a template and the large picture. The computation of the correlation formula is done via a fast fourier transform. The maximum of the matrix gives the row and column coordinates of the translational alignment. Hall et al [21] apply a coarse scan by shifting the window one half its height or width over the larger picture. A fine scan at promising points then gives the proper match point. It was found by the present author, that shifting by as much as half of the window's height or width is really too much and at times match points could not be located.

3.3 Automatic Selection of Control Points

In the control point approach, control points are usually selected manually by looking at the photographs or line printer output maps of the pictures. Sharp corners or

regions with high discontinuity levels in the gray levels of the picture present promising control points. Automatic selection of control points implies that the picture is scanned and an ordered list of control points is produced. Steiner [58] and Fleming [18] punch holes at control point locations in photographs, which are then digitized. The process is tedious, inefficient and inaccurate. Bernstein [8,9] has attempted to automatically locate pre-specified control points in pictures. It is important to note that these control points were either fixed reseau marks or ground control points embedded in the pictures. The detection of reseau marks is simplified, since their size and shape is fixed, and hence a fixed template can be used.

Scott [55] has also recently attempted to automatically select control points. A number of closed features such as lakes are located in the pictures. To identify these selected lakes, feature extraction parameters such as area, length and angle of orthogonal axes, mean and variance of the lakes are computed. These parameters are then used for recognition purposes. The proposed technique is only applicable to pictures where the number of lakes is sufficiently high. In general, automatic control point selection should be independent of any pre-specified objects in the pictures.

In this section, an attempt is made to <u>automatically select control points</u>. Thus a fixed template approach cannot be followed, because the technique developed should

be applicable to any picture. For a restricted picture set, e.g., pictures of agricultural areas, where regular boundaries are encountered, control point selection is considerably easier than in pictures with either uniform or gradually changing regions, e.g., naturally forested regions. In order to automatically choose control points, two basic points should be taken into consideration, namely:

1. What should be the features of a good control point?
2. What spatial distribution (or alignment) of control points produces the most accurate registration?

The following paragraphs will discuss these two points in detail.

3.3.1 Features of a Good Control Point

Control point selection should be independent of the specific gray levels in the picture, and a control point should be easily identifiable, both visually and algorithmically. It may be the intersection of sharp lines or discontinuities in two dimensions, which should be orthogonal. In an ideal region, i.e., with no noise, a single pixel with maximum gray level discontinuity with its background is undoubtedly the best control point. The single pixel as a control point will not have any scaling or rotational problems; however, in practical situations matching a single point of the first picture with the second picture will lead to many false matches. Thus some area around the control point with unique information should be

considered, e.g., intensity or structural information. The procedure should reject:

1. Linear features, where the possibility of multiple matches may occur in the second picture (during location of the same control point).
2. Uniform regions, because ambiguity may result.
3. Objects which have undergone changes, because the registration procedure will produce incorrect results.

If the control point is to be independent of actual gray levels, one possible technique for its selection is to obtain a binary gradient picture. Because of noise problems, points in the picture where the gradient is less than or equal to some threshold T, are assigned zero values. In order to obtain the control point from the binary gradient picture, two measures will be considered.

In the first measure, the number of non-zero elements in the window of the binary gradient matrix is computed. The motivation of this simple and fast technique is to be able to reject semi-uniform regions and linear features. A semi-uniform region will give a small edge, resulting in a smaller number of non-zero elements; however, a good control point may also get rejected. The second measure is the number of connected edge elements in the window of the binary gradient picture, see Chapter II. If the number of connected pixels is high, a potential control point may exist. This technique will reject noisy regions, since they produce isolated edge points. It should be pointed out that

this measure may not discriminate against straight edges. The rotational information from the gradient operator can be used to reject straight edges.

The selection of control points will be discussed for a 256*256 picture. For larger pictures, such as LANDSAT, the control points can be selected from sub-pictures of maximum size 256*256. The main advantage is in the saving of computer input/output time, as the larger picture need not be completely read into computer memory. Further comments on processing large pictures are given in later paragraphs.

Suppose the binary gradient picture of size 256*256, is divided into sub-regions of size 32*32. In each sub-region, one potential control point will be chosen. The total number of sub-regions is 64, but the 15 sub-regions of the last 32 rows and last 32 columns are different, since a 16*16 window cannot be displaced more than 16 units. A window is placed in the top left corner of the sub-region. A figure of merit using either of the two measures is calculated. The window is then displaced to the right and the figure of merit is calculated again. The maximum value of the figure of merit for all displacements in both directions gives the location of a potential control point. In order to speed computation, the window is displaced every second row and column. It should be noted that the window can span other sub-regions, so that potential control points at the boundaries of sub-regions will also be considered.

Also, the size of the sub-regions and the windows can vary. The processing of 49 sub-regions gives the locations of 49 potential control points.

3.3.2 Spatial Distribution of Control Points

Once enough potential control points with desired features have been selected, the final set of control points should be chosen such that:

1. They are evenly distributed over the picture.
2. Separation between points is a maximum.

The spatial distribution is achieved by linear ordering the 64 points into an 8*8 matrix representation, i.e., the matrix as given in Table XI contains the control point numbers 1,2,...,8 for the first row, and 9,10,...,16 for the second row, and so on.

Table XI. Spatial Distribution of 64 Points.

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

Next, the 49 potential control points corresponding to

the first 7 rows and 7 columns in Table XI, are grouped into 9 sets, the mapping of which is given in Table XII. The choice of the mapping varies, and the one which is given in Table XII, is one of the many selections. The 9 sets of points are represented by a 3*3 potential control matrix, as shown in Fig. 26. Each number in this matrix represents the number of potential control points chosen from the corresponding spatial coordinates, e.g., the number 4 in the top left corner represents 4 potential control points within the region, which represent the potential control point numbers 1,2,9, and 10.

Table XII. Mapping of Control Point Numbers.

Number	Set of Potential Control Points	Total Control Points
1.	(1,2,9,10)	4
2.	(3,4,5,11,12,13)	6
3.	(6,7,14,15)	4
4.	(17,18,25,26,33,34)	6
5.	(19,20,21,27,28,29, 35,36,37)	9
6.	(22,23,30,31,38,39)	6
7.	(41,42,49,50)	4
8.	(43,44,45,51,52,53)	6
9.	(46,47,54,55)	4

For choosing the 9 final control points from 49 potential control points, the potential control point with a maximum figure of merit in each group is assigned as the

4	6	4
6	9	6
4	6	4

Fig. 26. Forty nine potential control points.

control point, as shown in Fig. 27, where the numbers indicate the control point number. If two control points are very close, i.e., within ±5 pixels, the control point with the maximum figure of merit should be selected.

1	2	3
4	5	6
7	8	9

Fig. 27. Control point numbers.

The selection of control points as discussed above was restricted to a 256*256 picture. If the picture size is larger, then the picture can be divided into sub-pictures of a maximum size of 256*256. Potential control points for all the sub-pictures can be obtained and then the desired number of control points may be selected from them. Alternatively,

if a small number is desired, then 9 control points from each sub-picture may be obtained, and the desired number can be selected from the total obtained for the complete picture.

Once a set of control points has been chosen, it is important to check for their validity. For example, an object represented by a control point may not be included in the second picture or some change may have occurred in the object. To check for validity, a window around the chosen control point is extracted. This window is then cross correlated with the second picture for obtaining the approximate displacement. The maximum value of the two dimensional cross correlation gives the best match point. It is important to note that the following problems can occur:

1. The correlation value may be low, because of uniform regions. A control point may be defined as valid, if the correlation value is greater than 0.5.
2. For objects which have undergone changes, the correlation value may not be accurate. It is difficult to know automatically whether the corresponding control point in the second picture has changed. If a control point is chosen from such a region, registration accuracy will suffer.
3. Multiple matches may occur and the maximum value may indicate the location of a wrong point. This can be detected by checking for correlation peaks, which have

approximately the same value.

4. The object from which a control point is chosen, may not be included in the second picture.
5. If scaling and rotation parameters are large, the correlation technique may not be reliable in identifying corresponding control points. Thus the proposed technique can be applied to pictures, where scaling and rotational parameters do not vary by too great an amount.

The proposed technique for automatic selection of control points can be formally summarized in the following algorithm:

Algorithm C:

Step 1: Obtain the binary gradient picture with threshold T, from the original picture of size 256*256. For a large picture, divide the picture into sub-pictures of maximum size 256*256, and apply this algorithm on each of the sub-pictures.

Step 2: Divide the binary gradient picture into 64 sub-regions of size 32*32, by dividing the row and column dimensions by 8. The 15 sub-regions of the last 32 rows and last 32 columns regions are not considered.

Step 3: For a sub-region, place a 16*16 window at the top left corner of the sub-region and obtain the figure of merit, by using the connected edge

measure. Shift the window to the right and calculate the figure of merit again. Repeat this step for all points in the sub-region. To speed up the computations, rows and columns can be skipped. The maximum of the figure of merit gives the coordinates of the potential control point. Repeat this step for all the 49 sub-regions, producing a list of 49 points.

Step 4: By using mapping as given in the Table XII of the contrcl point numbers, group the numbers into 9 spatially distributed sets.

Step 5: From the 9 sets, as given in the potential control pcint matrix, assign the potential control point with the largest figure of merit in the group to the final control point.

Step 6: Check the validity of the control points.

In order to check the performance of the algorithm, two different sets of pictures were chosen for testing. In the first case, two sub-pictures from LANDSAT images 1344-18071 (July 1973) and 1160-17553 (July 1975) were extracted. The size of the pictures is 600*600 and they are shown in Fig's 28 (Edmonton I, band 5) and 29 (Edmonton II, band 5). The Edmonton I picture was extracted from the 1344-18071 image with coordinates (1416,2015,1888,2487), where the notation (, , ,) represents first row, last row, first column, and last column respectively. The Edmonton II

Fig. 28. Edmonton I with control points.

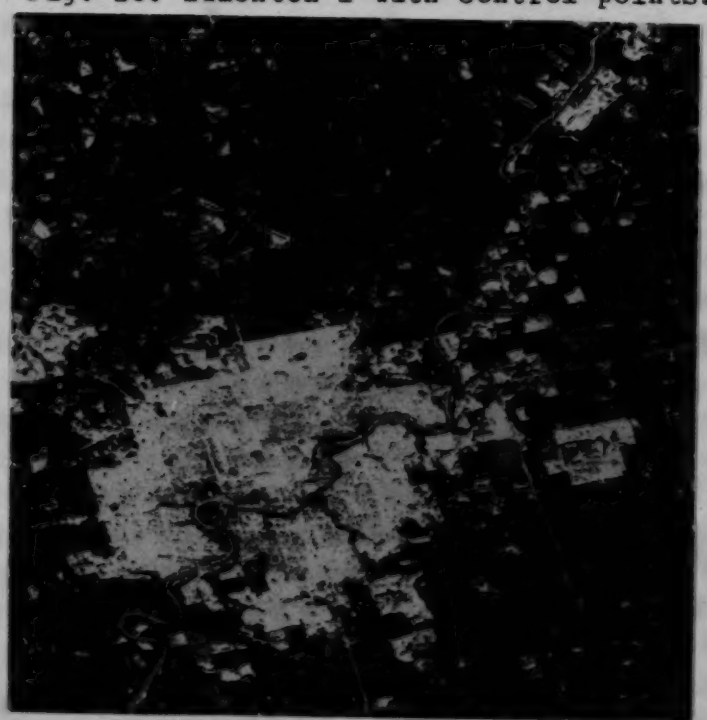

Fig. 29. Edmonton II with control points.

picture was obtained from the 1160-17553 image with coordinates (1687,2286,1819,2418).

The binary gradient picture of the Edmonton II picture was obtained with threshold set to 2. The binary gradient picture (512*512) of Edmonton I picture with threshold set to 12 is shown in Fig. 30. The binary picture of Edmonton II was divided into sub-pictures of maximum size 256*256, resulting in 4 sub-pictures of size 256*256, 2 sub-pictures of size 256*88, 2 sub-pictures of size 88*256 and a 88*88 sub-picture. The algorithm produced lists of 49 potential control points for each sub-picture of size 256*256 and the set of 9 control points were selected for each such sub-picture. For sub-pictures smaller than 256*256, the number of potential control points were less than 49 and the set of control points were also less than 9. Next, the control points which were not included in the second picture were deleted and a set consisting of the 9 best control points was selected.

For checking the validity of the control points, windows of size 16*16 from the top left corner of the control points were extracted. The dimensions of the window were chosen such that they are optimum. For small dimensions, the correlation value may not be reliable and for large dimensions, the computational time will be excessive. These windows were then positioned with respect to the corresponding control points of the second picture.

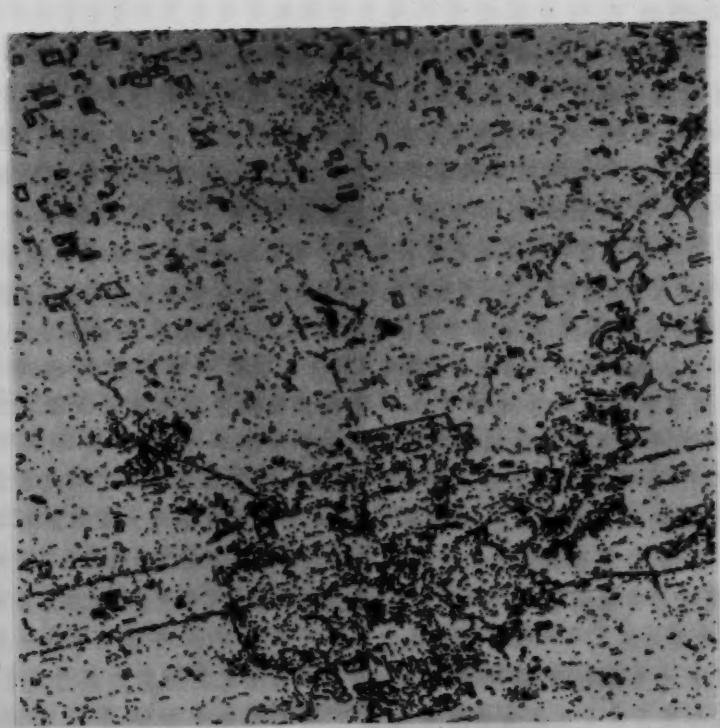

Fig. 30. Binary gradient picture of Edmonton I.

A two dimensional cross correlation matrix was computed by shifting the windows horizontally and vertically for a displacement of 8 units. The maximum correlation value in the matrix gave the location of best matching integer coordinates. The correlation procedure was repeated for getting fractional alignments of one half pixel.

For control points from either uniform regions or regions where changes occurred in the second picture, correlation values were less than 0.5. Thus 3 control points were dropped. Both measures of calculating the figure of merit were used. The second approach of connected edge elements is more reliable, because it takes care of noisy regions and isolated edge pixels. The 6 control point areas are outlined in white squares in Fig's 28 and 29. The central pixel within the square represents the control point chosen.

In the second test case, two pictures of the fertilizer region were chosen. The pictures are 600*500 and are shown in Fig's 31 (Fertilizer 2, red band) and 32 (Fertilizer 3, red band). Both pictures do not have well distributed feature points, and have more than minor variations in rotation and translation. The binary gradient picture of the Fertilizer 2 picture was obtained with threshold set to 5. The binary picture was divided into sub-pictures of maximum size 256*256, resulting in 2 sub-pictures of 256*256, 2 sub-pictures of 256*244, 1 sub-picture of 88*256 and a sub-picture of 88*244.

Fig. 31. Fertilizer 2 with control points.

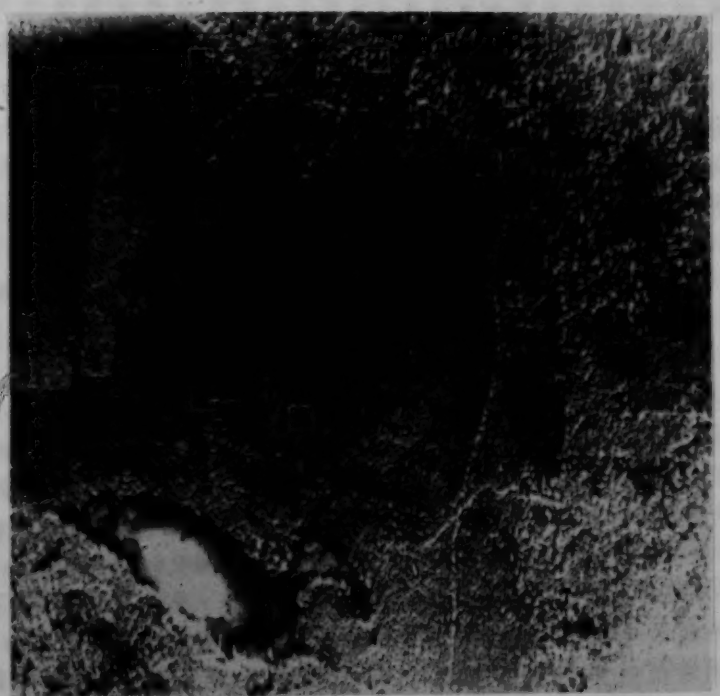

Fig. 32. Fertilizer 3 with control points.

The connected edge element technique was used to compute the lists of potential control points. A set of 9 final control points were chosen from the control points obtained over the whole picture.

The procedure followed in selecting the 9 control points and in checking their validity is same as in the first test case. The correct control points obtained are shown in Fig's 31 and 32. It is worth noting that except for one control point, all the rest were chosen at the edges of the objects of the picture. This particular control point was chosen to satisfy the criterion that at least one control point should come from the region in which no control point was selected. This problem will arise for uniform regions, since they will produce weak edges and consequently the connected edge element measure will have a low value.

3.4 Iterative Feedback Approach

As mentioned in section 3.2, if the control points are not selected properly, mis-registration occurs. For the improperly selected control points, the error was found to be greater than 5 pixels. In order to handle this problem, an iterative feedback technique for registration was developed.

The proposed approach is a combination of the control point and correlation approach. As in the control point technique, some points are chosen and coefficients of a

transformation F are calculated. Next, a small window corresponding to a control point is extracted and its corresponding transformed window (larger in size) is also obtained. These two windows are then cross correlated for a shift of ±2 units in both directions. The small shift range is useful for convergence in the next iterative loop, as the shift values are used in feedback correction. The two dimensional displacement vector is computed for all the control points, and computation of the correlation matrix is repeated for obtaining fractional alignments in terms of one half pixel. Next, a feedback correction of the displacements is done for all the control points of the original picture.

The whole process of computing the coefficients of the transformation, extracting windows and computing the two dimensional displacement vector is repeated until either convergence is attained or a maximum number of iterations has been reached. Convergence is defined to imply that the displacement vector is zero for all control points. For a non-zero displacement vector, the best set of control points is chosen from the minimum displacement vector. The procedure tries to adjust the control points of the original picture to give the best possible coordinates. The transformation is then generated using the minimum displacement vector. As a final check, the original picture and the transformed picture are cross correlated at both specific and random test points. The proposed method is given in

algorithm D.

Algorithm D:

Step 1: Set max=1, maxiter: maximum number of iterations, p: shift parameter. Read control point pairs from the two pictures.

Step 2: Compute the coefficients of the transformation by least squares analysis for the control points.

Step 3: Extract a small window S of size w*w around one control point from the original picture.

Step 4: Calculate the larger transformed window WT of size w+p*w+p, corresponding to S, from the picture to be transformed.

Step 5: Cross correlate window S for a shift of p units with the transformed window WT, giving a two dimensional matrix, from which the relative displacement with respect to the control point of the original picture is calculated.

Step 6: Repeat steps 3 to 4 for all control points.

Step 7: max = max + 1, if max ≥ maxiter, go to step 8. If the relative displacement vector is zero in both directions, go to step 8. Add the relative displacement vector to the control points of the original picture in both the directions (x,y), and save the vector. Repeat the procedure beginning with step 2.

Step 8: Choose the control point pair, according to the minimum relative displacement vector obtained from

several iterations.

Step 9: Repeat step 2 with new control point pairs and transform the whole picture.

To illustrate the iteration procedure, suppose the control point coordinates of the original and second picture are (40,10) and (100,25) respectively. After extracting the window and transformed window and computing the cross correlation, let the relative displacement vector be 1 unit in the x direction and -2 units in the y direction. After applying the feedback correction, the new control point pair becomes (41,8) and (100,25).

It should be pointed out that the transformation function generates row and column coordinates as non-integer numbers. In the nearest neighbor approach, the numbers are rounded, i.e., 0.5 is added and integer portions are taken as coordinates. These fractions can also be used as weights in the two dimensional interpolation. If w1 and w2 represent fractions of coordinates generated, and a(x,y) is the gray level of the pixel at row x, column y, then the gray level of the output pixel is given by:

$$(w1*a(x,y) + (1-w1)*a(x+1,y) + w2*a(x,y+1) + (1-w2)*a(x+1,y+1))/2.$$

In order to test the performance of the algorithm, four different sets of pictures were chosen for

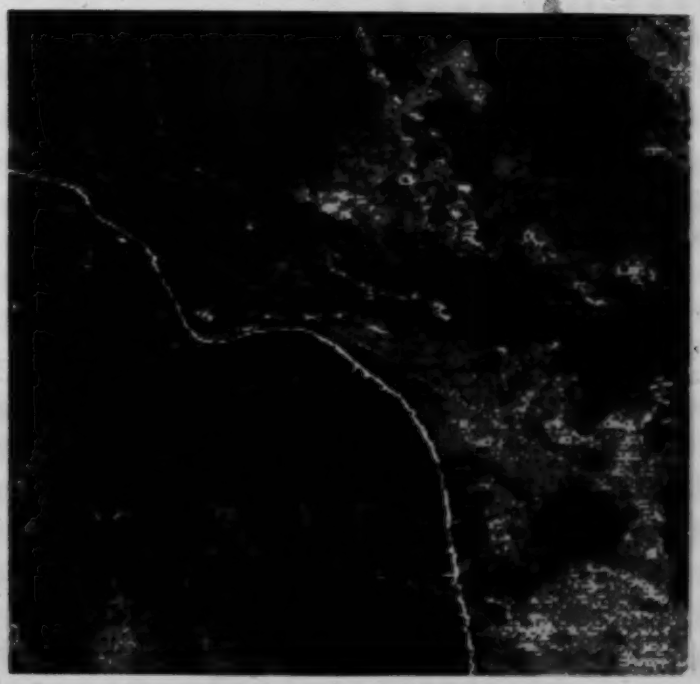

Fig. 33. Lake I.

Fig. 34. Lake II.

registration. It should be noted that the size of the registered pictures varied from 256*256 to 501*501. In the first case, two sub-pictures from LANDSAT images 1043-18341 (Sept. 1972) and 1600-18242 (March 1974) were extracted. The size of the pictures is 256*256 and 384*384 and they are shown in Fig's 33 (Lake I, band 5) and 34 (Lake II, band 5). The Lake I picture was obtained from image 1043-18341, with coordinates (2030, 2285, 2691, 2946). The Lake II picture was extracted from image 1600-18242, with coordinates (15,398,1536,1919). The window size around the control points was chosen as 9*9 with the maximum number of iterations set to 5. The registered picture of size 256*256, as shown in Fig. 35, was generated by a nearest neighbor approximation and by applying a first order transformation, whose coefficients are:

$a = 0.99 \quad b = -0.019 \quad c = 1.87,$

$d = 0.036 \quad e = 1.01 \quad f = -52.97.$

The corresponding windows of the two registered pictures were cross correlated. The final registration accuracy for 20 reference points was about 1 to 2 pixels. In the window matching, normalized correlation failed at those points where objects had undergone marked changes, even though objects in the two pictures were essentially the same, namely, a lake from fall and winter seasons. To correct this problem, instead of using normalized correlation, it is proposed that the edges of the two windows be obtained by a gradient operator. The two binary gradient

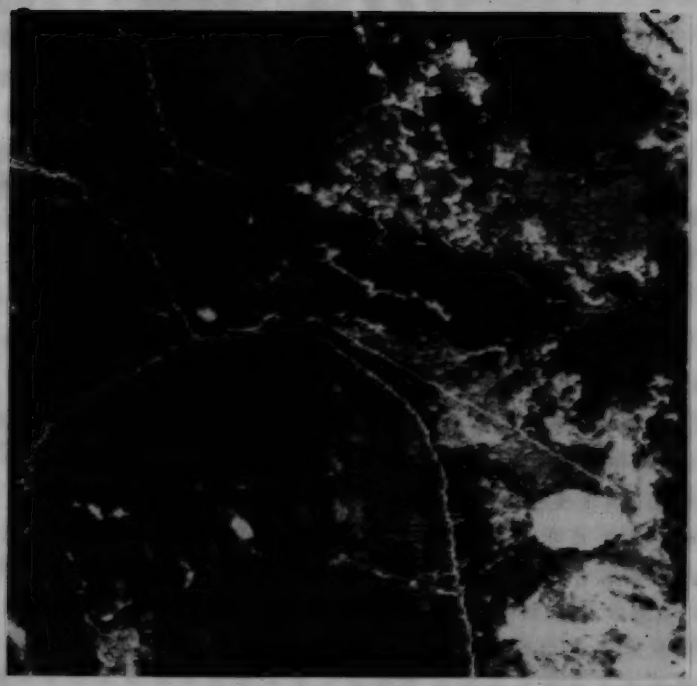

Fig. 35. Lake II registered with Lake I using nearest neighbor approximation.

Fig. 36. Fertilizer 3 registered with Fertilizer 2 using nearest neighbor approximation.

pictures can then be registered. Comparing with Jayroe et al's [27] method, it follows that there is no need to generate full binary gradient pictures, but only 2*N*I windows, where N is the number of control points, I is the number of iterations (maximum of 5). Thus this scheme of registration is much faster than Jayroe et al's.

In the second test case, the initial control points were obtained by the automatic selection technique. The pictures are given in Fig's 31 and 32. The window size around the control points was chosen to be 15*15, with the maximum number of iterations set to 5. The best set of control points was chosen for generating the registered sub-picture of size 420*350. The registered sub-picture, as shown in Fig. 36, has a registration accuracy of 1 pixel for the second order transformation. The accuracy was calculated by using the two dimensional cross correlation function at 20 specific and random test points. The coefficients of the second order transformation are:

a = -60.71 b = 0.89 c = 0.11 d = -0.7E-6 e = 0.6E-4,
f = -0.11E-3 g = 201.35 h = -0.11 i = 0.88 j = 0.4E-6,
k = -0.7E-4 m = 0.2E-4.

The registered picture was also generated by using a two dimensional polynomial function among the input pixels as shown in Fig. 37. Bilinear interpolation smooths pictures and takes care of discontinuities. The nearest neighbor and bilinear techniques were also applied to the green and blue bands of the color photograph, of the

fertilizer region pictures. The registration accuracy was within 1 pixel for all bands. It should be noted that the two dimensional polynomial function was also used in generating the windows for the iterative feedback process. The set of final control points is more accurate, and therefore, the transformation coefficients are not the same as obtained for the nearest neighbor approximation. The coefficients of the second order transformation are:

a = -61.5 b = 0.90 c = 0.10 d = -0.7E-6 e = 0.8E-4,
f = -0.12E-3 g = 201.7 h = -0.12 i = 0.89 j = 0.4E-6,
k = -0.11E-3 m = 0.4E-4.

The registered picture was also generated by using the cubic convolution function [47]. The gray level of the output pixel was computed by using the 16 gray levels of the input picture. The weights used were derived from the truncated sinx/x function. The coefficients of the second order transformation are:

a = -60.2 b = 0.9 c = 0.10 d = -0.14E-5 e = 0.7E-4,
f = -0.1E-3 g = 203.8 h = -0.13 i = 0.89 j = 0.9E-6,
k = -0.13E-3 m = 0.72E-4.

The registered picture is given in Fig. 38 and has a registration accuracy of about 1 pixel. The accuracy was checked by cross correlating corresponding windows as in the previous interpolation techniques.

The third test case is concerned with the registration of Edmonton I and Edmonton II pictures as shown in Fig's 28 and 29. Using the automatic control point approach, control

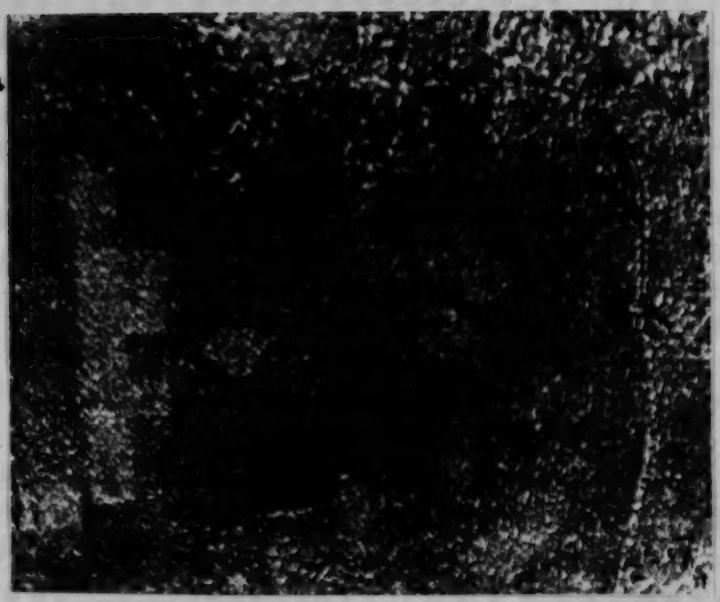

Fig. 37. Fertilizer 3 registered with Fertilizer 2 using bilinear approximation.

Fig. 38. Fertilizer 3 registered with Fertilizer 2 using cubic convolution.

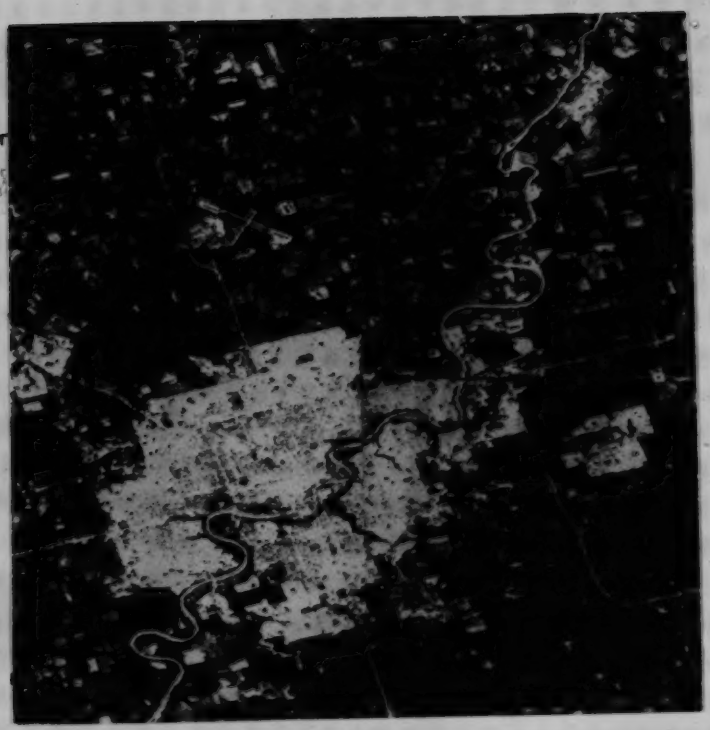

Fig. 39. Edmonton I registered with Edmonton II using nearest neighbor approximation.

points were obtained for the least squares analysis. The coefficients of the second order transformation are:

a = 61.20 b = 1.0 c = -0.012 d = 0.76E-7 e = 0.1E-5,
f = 0.3E-4 g = 64.73 h = 0.038 i = 1.03 j = -0.46E-7,
k = -0.15E-4 m = -0.14E-3.

The registered picture, given in Fig. 39, has a registration accuracy of about 1 pixel and was checked both visually on a line printer map and by correlation at 20 random points. As band to band variation is of interest in change detection, the other three bands of the two pictures were also registered. It should be noted that the same transformation function was used for obtaining registered pictures of all bands.

The final test case deals with the registration of forestry cut areas of northern Alberta. Digital data from LANDSAT images 1043-18341 (Sept. 1972) and 1439-18325 (Oct. 1973) were extracted. The pictures are 512*512, but only 256*490 windows are shown in Fig's 40 (Cut I, band 5) and 41 (Cut II, band 5). Cut I picture was extracted from the 1043-18341 image, with coordinates (256,767,1731,2242). Cut II picture was obtained from the 1439-18325 image, with coordinates (200,711,1880,2391). The control points were selected automatically and corrected by the iterative feedback approach. The coefficients of the second order transformation are:

a = 134.8 b = 1.0 c = 0.021 d = -0.32E-6 e = -0.25E-4,
f = -0.65E-4 g = 11.31 h = 0.17E-1 i = 1.01 j = -0.15E-6,

Fig. 40. Forestry Cut I.

Fig. 41. Forestry Cut II.

Fig. 42. Forestry Cut I registered with Forestry Cut II using nearest neighbor approximation.

k = 0.97E-5 m = -0.1E-3.

The registered picture shown in Fig. 42 was generated by applying the transformation to the picture given in Fig. 40. The registration accuracy was about 1 pixel and was checked visually and by doing correlation at 20 selected and random points. The automatic control point procedure gave a list of 196 potential control points. 90 control points were not considered since they were missing in the second picture. The root mean square error for the 70 control points was 0.8 pixels. The 20 control points with low correlation values were not taken into consideration.

3.5 Registration with Large Scale Differences

This section deals with the problem of registration of pictures which have been obtained by different methods, e.g., aircraft and LANDSAT. Previous registration techniques have limited the scale factors to be about 1:1.5. In surveying registration techniques, it was noticed that alignment of pictures with scale differences as large as 1:5, had not been reported. A technique is presented to solve this problem and is tested on digital pictures with scale differnces of about 1:6 (horizontally) and 1:4.2 (vertically). Since the first LANDSAT satellite was launched in 1972, the proposed technique is very useful for detecting changes before 1972 (assuming aircraft data exists).

In the standard nearest neighbor approach, the gray

level corresponding to the rounded address from the large picture is used for the registered picture. If this approach is used for registration, then the registered picture will be degraded. For example, if the scale factor is 1:b, then this technique chooses one pixel from the b*b neighborhood. If b is greater than 2 and if the neighborhood contains edge transitions, a pixel corresponding to either high or low contrast can be chosen. This causes the straight edges in the registered picture to look irregular, and non-noisy regions to appear noisy.

In the averaging approach, by knowing the scale factors, the larger picture can be averaged to reduce the scale and then the two pictures can be registered. If this is done, registration may not be accurate. The reason for this is that averaging changes the relative coordinates and size of the objects in the picture. Also, because of averaging, the registered picture will be smoothed. The larger picture can also be sampled to reduce its size, but the reduced picture will still be degraded.

In order to take care of these problems, an improved technique is presented as follows: Suppose I is the row number of the registered picture to be generated, then the row and column addresses for rows I-1, I, and I+1 are generated as shown in Fig. 43. Let the column under consideration be J, i.e., the output pixel is located at row I and column J. The registered picture is generated row by row. The advantage is in the saving of computer storage,

since the registered picture need not be kept in core.

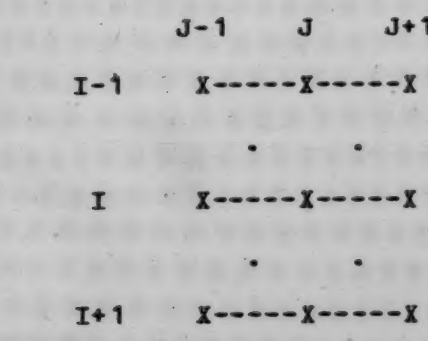

Fig. 43. Rectangular region.

In the proposed technique, the equations of four lines which are midway between (I-1,I), (I,I+1), (J-1,J) and (J,J+1) are calculated. These lines define a rectangular region, the coordinates of which are shown as four dots in Fig. 43. The output pixel is assigned the average gray level of the region defined by the four dots, whose shape also varies with I and J. Since the boundary of this region is not aligned with the grid, two approaches for defining the area within the region exist, which are:

1. Zeroth order approach. In this approach, the addresses of the pixels along the boundary are rounded, i.e., a pixel is included or not depending on whether or not its coordinates are inside or outside the boundary. This approach is simple to use but may not be very accurate.
2. Refined approach. In this approach, the pixel's contribution to the region, as defined by the boundary line, is calculated, i.e., the gray level of the

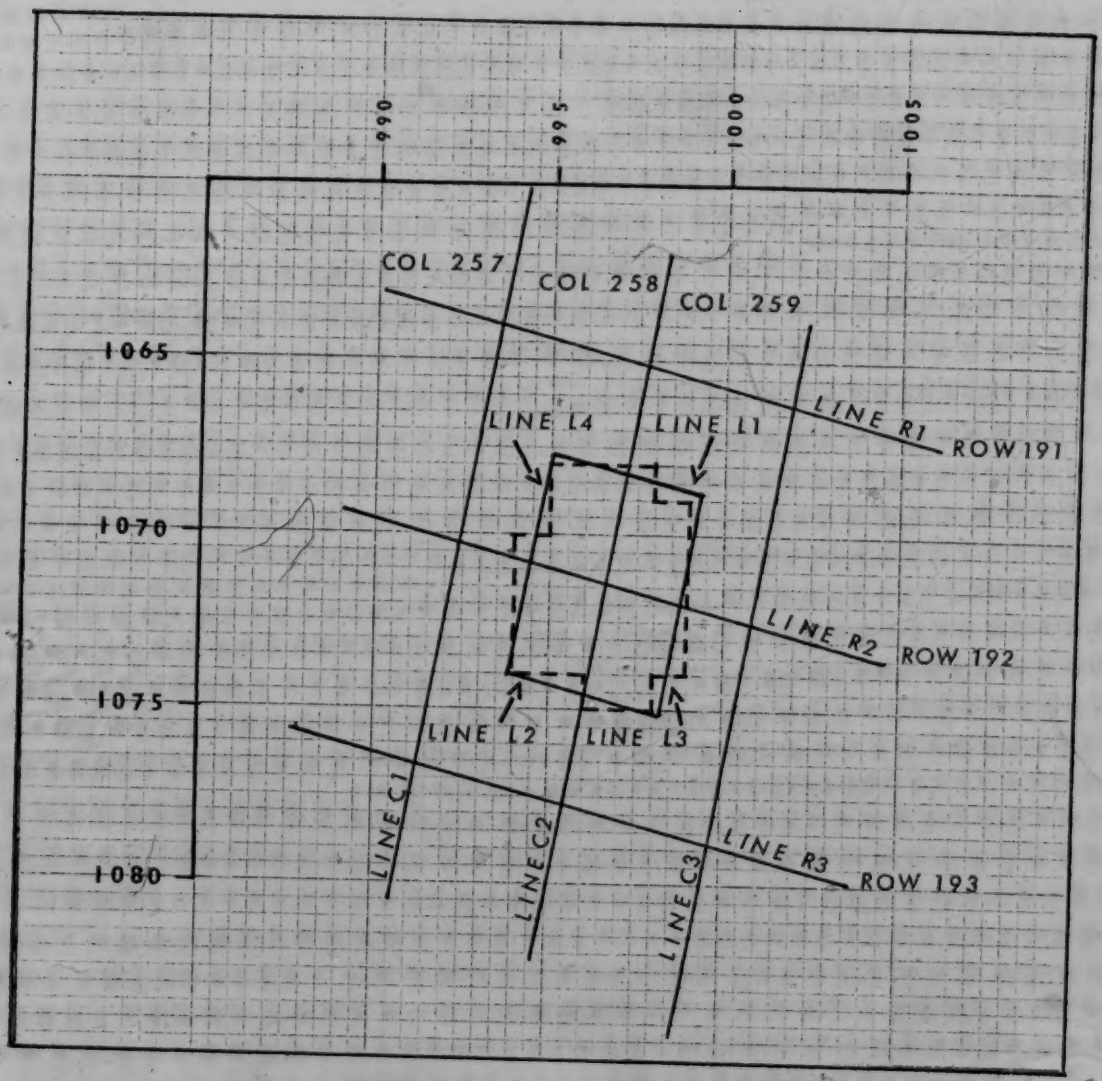

Fig. 44. Zeroth order approximation to the rectangular region.

boundary pixel is proportional to the area occupied by the pixel within the boundary line. This approach is very accurate; however, it is computationally expensive.

In order to compare the nearest neighbor and the proposed approximation, processing for one output pixel will be discussed in detail. For the output pixel corresponding to the row 192 and column 258, the address generated is (1071.4,996.6), see Fig. 44. The gray level corresponding to the address (1071,997) of the picture will be used for the nearest neighbor approximation. It is evident from Fig. 44 that this approximation is not only very poor but also does not use any information from the pixel's neighborhood. Also, due to large scale differences, the distance between lines R1, R2, R3 and C1, C2, C3 connecting the input pixels is large. In fact, the difference is about 5.99 pixels horizontally and about 4.26 pixels vertically.

In the proposed approximation, the equations of lines L1, L2, L3, and L4 which are midway between lines (R1,R2), (R2,R3), (C2,C3) and (C1,C2) respectively are calculated. These four lines define a rectangular region, which is the scaled down version of the region defined by lines R1, R3, C1, and C3. The area within this region is of interest for computing the average gray level for the input pixel with coordinates (1071.4,996.6). Since the boundary of this region is not aligned with the grid, the zeroth order approach along the boundary is taken. The dotted line in Fig. 44 represents the approximation to the rectangular

region. The average of the gray levels of the region defined by the dotted lines is assigned as the gray level for the input pixel with coordinates (1071.4,996.6). It should again be noted that the shape of this approximation is not fixed but varies from pixel to pixel.

The test problem is concerned with registration of an aerial picture of Edmonton (1971) with a LANDSAT picture (1975) of size 600*600, extracted from 1160-17553 image. The LANDSAT picture as shown in Fig. 45 is of size 281*400 (window of Fig. 29) and the aerial picture's size is 2048*2048. A 400*256 window of the aerial picture is shown in Fig. 47. In order to compare the proposed technique with the averaging approach, the aerial picture was averaged as shown in Fig. 46, i.e., from a 4*4 neighborhood was averaged to produce the size reduction. Control points in both pictures were selected manually and for the nearest neighbor and the proposed technique, control points were obtained from their original pictures, i.e., from the aerial (2048*2048) and the Edmonton II (600*600). The registered pictures as shown in Fig's 48-50, were generated by a second order transformation, whose coefficients for the averaging approach are:

a = -94.12 b = 1.46 c = 0.38 d = 0.4E-6 e = -0.35E-3,
f = 0.14E-3 g = 12.45 h = +0.167 i = 1.092 j = -0.6E-6,
k = -0.39E-3 m = 0.9E-4.

The transformation coefficients for the nearest neighbor and proposed approximation are:

Fig. 45. Window of Edmonton II.

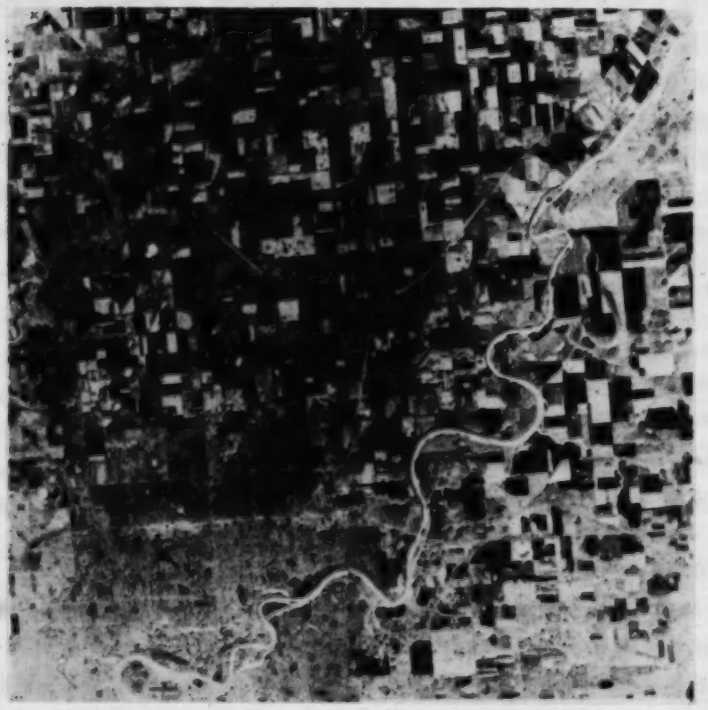

Fig. 46. 4*4 average of aerial picture.

Fig. 47. Window of aerial picture.

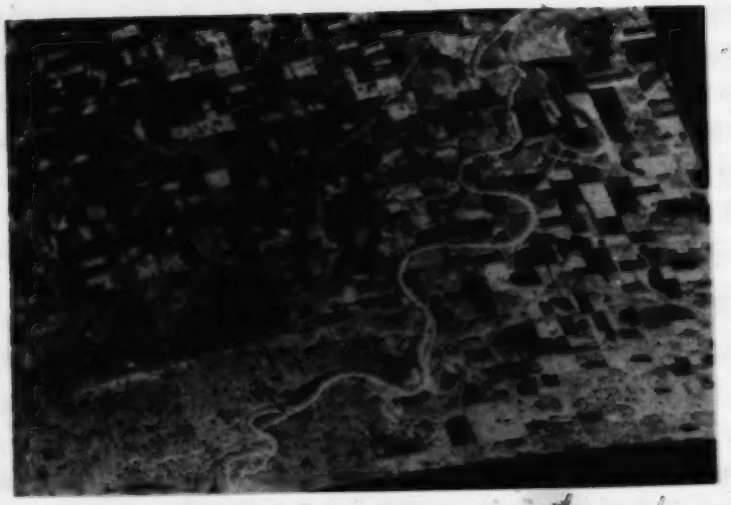

Fig. 48. Average aerial picture registered with Edmonton II using nearest neighbor approximation.

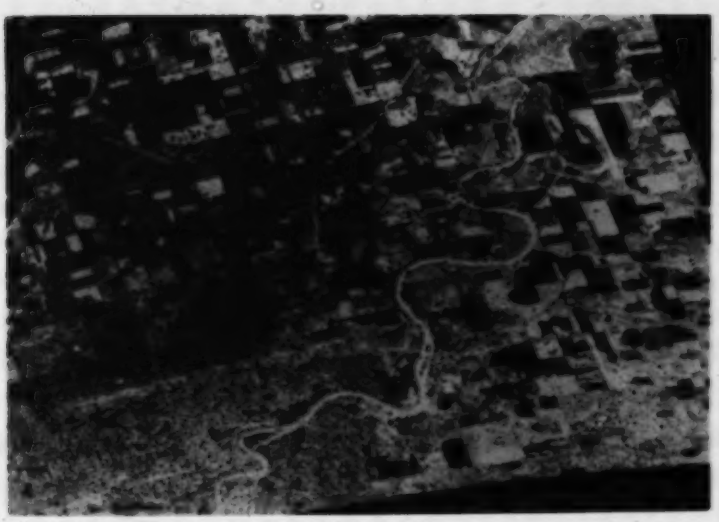

Fig. 49. Aerial picture registered with Edmonton II using nearest neighbor approximation.

Fig. 50. Aerial picture registered with Edmonton II using proposed approximation.

a = -376.51 b = 5.84 c = 1.52 d = 0.16E-5 e = -0.14E-2,
f = 0.57E-3 g = 49.80 h = -0.66 i = 4.36 j = -0.2E-5,
k = 0.38E-3 m = -0.15E-2.

The nearest neighbor picture was generated by using the gray level of the pixel, which is closest to the calculated row and column addresses of the original aerial picture. In the proposed technique, the zeroth order approach was used in calculating the rectangular region's average gray level, see Fig. 44 for a particular output pixel. Clearly, this picture is the best visually by comparing with the other two registered pictures. In order to check registration accuracy, the correlation approach cannot be followed, because of severe changes. Visually, the registration accuracy was found to be reasonable, i.e., within 1 to 2 pixels. The computational processing time for registration was high for both the nearest neighbor and the proposed technique. The equations of lines need not be generated for every output pixel, i.e., for each output row, equations of lines L1 and L2 will not change, and only line L3 needs to be recomputed.

3.6 Discussion of Results

This chapter deals with registration of pictures. In the automatic control point selection, some basic features of a control point and problems encountered in selecting them are given. Control points are selected on the edges of the objects in the pictures. The technique is independent

of gray levels in the pictures and does not use any pre-specified objects or templates.

In the iterative feedback approach, the control points are corrected to produce the best set of points. The registration accuracy was between 1 to 2 pixels for the first order transformation and 1 pixel for the second order transformation. The accuracy was computed by cross correlating corresponding windows at 20 reference points. The root mean square error was 0.8 pixels and the size of registered pictures varied from 256*256 to 501*501.

In the registration of pictures with large scale differences, a new technique is presented. The row and column coordinates of the input picture for the three output rows, i.e., previous, current, and the next are calculated. These coordinates define a region, the shape of which shows the scale differences. In the nearest neigbor approach for the output pixel, only 1 pixel is used from this region. If the size of the region is large, most of the important information about the region is lost. In the new technique, 4 boundary lines define a smaller version of the region. Since the output picture's grid is not aligned with the input picture's grid, a zeroth order approach along the boundary is taken. The average gray level of the new region is computed and used for the output pixel. The technique is compared with the nearest neighbor and the averaging approaches. The picture generated by the proposed technique is the best visually, as objects look smooth and non-noisy.

Chapter IV

TEMPLATE MATCHING

Template matching is a basic technique of pattern recognition, which has been used extensively in character recognition. Templates from a data base are compared with the incoming patterns and a correlation coefficient is calculated for each template. The template with the greatest resemblance to the incoming pattern is the one having the maximum correlation coefficient. In digital picture matching, for finding an object or its multiple copies, a template or window is displaced over all possible points of the picture. In this chapter, methods for the movement of the template across the picture and for comparing with the sub-pictures will be examined. Various search techniques for moving a template include, sampling at certain points [21,22], and planning [30].

If correlation is negative, full computation is still done for all the corresponding rows of windows and sub-pictures. Thus Barnea and Silverman's [6] observation that computations need not be done with high accuracy at every reference point is very important. In their approach, random samples from the sub-picture and a window are taken and a difference sum is continuously monitored. If that sum

exceeds some pre-specified threshold, the reference point is dropped from further investigation.

One of the main problems in using a difference measure is that the mean of the window and means of the reference sub-pictures have to be computed. Also, the difference measure does not take care of the constant offset problem. Moreover, the threshold has to be specified. Barnea and Silverman do not suggest any method for choosing the pre-specified threshold; however, recently Onoe and Saito [41] have proposed a technique for automatic threshold setting.

Tasto and Block [60] describe a template matching method for detecting objects of known shapes, size and orientation. The template and the picture are reduced to binary images by gradient methods, to produce a contour map and a line drawing. The template is moved over the picture and the city block distance

$$d\{(i,j);(h,k)\} = abs(i-h)+abs(j-k),$$

is computed. The authors also give a reformulation of computational equations for efficiency. Rosenfeld and Pfaltz [52] suggest using the following measure:

$$d\{(i,j);(h,k)\} = \max[(2*abs(i-h)+abs(j-k))/3, \max\{(abs(i-h), abs(j-k))\}].$$

Nagel and Rosenfeld [39] sort the gray levels of a window in increasing order and note their coordinates. The same procedure is also used for the sub-pictures. The authors report that the absolute difference measure sum grows much faster for a non-match point. This technique is

shown by the authors to be faster than the row by row correlation. The basic concept is the same as that of Barnea and Silverman.

4.1 The Proposed Method

It is assumed that the object to be located is present in the picture. To take care of constant gain and offset, normalized correlation will be used as a similarity measure. The correlation between a window and a sub-picture is calculated by using all the rows; however, intermediate results can give an insight into the trend of the sign or value of the correlation coefficient. If the value is either negative or small, then the reference point can be dropped. The concept of a partial correlation coefficient is defined as:

Definition 5: A partial correlation coefficient is an intermediate correlation coefficient, when successive rows of a window and sub-picture are correlated. If the row size of a window is N, then a maximum of N partial correlation coefficients exist.

Let the window size be 4*4, then while matching, 4 partial correlation coefficients are computed for each of the 4 rows. In other words, the first coefficient is computed by matching the first row of a window with the first row of the sub-picture. The second coefficient is computed by matching the corresponding second rows and updating the first coefficient. It should be noted that the

value of the Nth partial correlation coefficient is same as the correlation coefficient.

The proposed method will take the following points into consideration:

1. For matching a window with sub-pictures, not all the rows need to be correlated for a non-match point. This point is implemented by monitoring the partial correlation coefficient at three check points.
2. The unit shift displacement of a window can be speeded up in both directions, i.e., a number of rows and columns of reference points can be skipped. For one approach, alternate rows and columns are evaluated to check whether the correlation is above some threshold.
3. While correlating rows, the order in which rows are correlated can be changed.

Let k1,k2,k3 be pre-specified constants. The range of these constants is between -1 and +1. For an M*M window, the partial correlation coefficient is computed and the value is monitored at three check points, namely M/4, M/2 and 3M/4. If the value of the coefficient is $\leq$ k1, when M/4 rows have been correlated, then computation is stopped for that particular reference point. If not, computation is carried on until M/2 rows have been processed. If the value of the coefficient is $\leq$ k2, the reference point is dropped, otherwise computation is continued until 3M/4 rows have been processed. At the third check point, 3M/4, if the partial correlation coefficient is $\leq$ k3, the reference point is

skipped, otherwise computation is continued to M rows.

For skipping column reference points, the correlation is computed at the first reference point. If the correlation value is ≤ k3, the next column point is skipped. Processing resumes at the third column point. For example, for columns 1,2,3,4,5,...,M, points 3,5,..., are evaluated, if points 2,4,..., were ≤ k3. Thus the first point is computed, and if the value is ≤ k3, the second point is skipped. The third point is then computed and so on.

For skipping complete rows of reference points, the first row is evaluated. If correlation values for all the row elements are ≤ k3, the evaluation of the next row is not done. Processing resumes with the third row. The motivation for skipping rows and columns of reference points and of monitoring correlation values at check points, is to decrease the computational time.

It was assumed that in matching, the first row of the two pictures was correlated, then the second row was correlated and so on. The ordering of rows can be changed by using a random number generator to generate a non-repeating integer sequence of numbers between 1 to M. Then, rows can be correlated according to the ordering of the integer numbers.

A general FORTRAN program was written to implement the proposed method, Barnea and Silverman's method, and conventional correlation. A further option could choose the pre-specified or random ordering of rows for matching. The fast

method with random row ordering will be called Fast Random. Barnea and Silverman's method was implemented with Onoe and Saito's [41] automatic threshold setting technique. The absolute difference measure and normalized absolute difference measure were used as similarity measures for the Barnea and Silverman's method. For the normalized absolute difference measure, means of the template and sub-pictures were subtracted before correlating them. The normalized absolute difference measure with random row ordering will be called as NADM Random. The values of k1,k2,k3 for the test problems were taken to be 0.0, 0.2, and 0.3 respectively.

In the first test problem, LANDSAT pictures from two different years were registered, using the iterative feedback correction technique, see Section 3.4. The two registered pictures are given in Fig. 33 (original) and Fig. 35 (registered) respectively. A 32*32 window was extracted from the coordinates (112,64) of the original picture. The problem was set to locate this window in the second picture. The range of search coordinates was limited to (100,30), (100,79), (149,30) and (149,79). The total number of reference points is 50*50=2500. For a 32*32 window, the total number of rows to be correlated is 32*2500=80,000. Details of using six matching techniques are given in Table XIII. It should be noted that CPU time is the total computational time to read in the pictures, complete the processing, search for minimum or maximum, and print the results.

Table XIII. Computational Times for Correlation Methods I.

Method	Rows correlated	Saving %	CPU secs	Ratio	Match point
1. Fast	8455	89.4	5.942	1.00	(112,64)
2. Fast Random	10086	87.4	11.267	1.89	(112,64)
3. Barnea with abs. diff. measure	59218	27.5	19.628	3.30	(101,30)
4. Barnea with normalized abs. diff. measure	56229	31.2	26.318	4.42	(138,56)
5. NADM Random	57939	29.1	39.222	6.60	(138,56)
6. Conventional	80000	0.0	27.158	4.57	(112,64)

In the second test problem, two pictures from a LANDSAT image were extracted. The 91*96 pictures are from band 6 and band 7 of image 1600-18242. The pictures were extracted with coordinates (181,271,1744,1839). The pictures correspond to the lake area, as shown for band 5 in Fig. 33. A 32*32 window corresponding to the lake in the band 6 picture was used as a template. The coordinates of the top left corner are (23,31). The range of search coordinates was limited to (20,20), (20,59), (59,20) and (59,59). The total number of reference points is 40*40=1600 and the total number of rows to be correlated is 32*1600=51,200. All six matching techniques were applied

and the results are given in Table XIV. The proposed method was also used to locate control points for 3 test pictures.

Table XIV. Computational Times for Correlation Methods II.

Method	Rows correlated	Saving %	CPU secs	Ratio	Match point
1. Fast	10730	79.0	3.862	1.00	(23,31)
2. Fast Random	12009	76.5	8.867	2.29	(23,31)
3. Barnea with abs. diff. measure	28773	45.3	8.639	2.23	(23,31)
4. Barnea with normalized abs. diff. measure	17686	66.9	10.605	2.74	(23,31)
5. NADM Random	16668	68.9	19.088	4.94	(23,31)
6. Conventional	51200	0.0	16.123	4.17	(23,31)

For the first test problem as summarized in Table XIII, it follows that the proposed method is the best among all the other methods. Barnea and Silverman's method failed to locate the proper match point. The two different measures gave different incorrect matching points. By applying random row ordering, the proposed method and Barnea and Silverman's method took more computational time. Again, Barnea and Silverman's method did not find the correct match point. As expected, conventional correlation by taking more time found the correct match point. The surprising thing is

that Barnea and Silverman's normalized measure took about same time as conventional correlation for this example. One of the reasons is that sub-picture means have to be subtracted at every reference point.

In the second test problem, as summarized in Table XIV, all six methods obtained the correct matching point. The proposed method is again the best. For random ordering, computational time was high with no apparent gain. In order to summarize, the following conclusions are drawn:

1. The proposed method is the best among the six tested. This method is applicable to any type of pictures, as constant gain and offset problems are taken care of.
2. Random row ordering should not be done. It consumed more computational time, with no apparent gain.
3. Barnea and Silverman's method should be used very carefully and only for those pictures where constant gain and offset problems do not arise.
4. The proposed method is 2 times faster than Barnea and Silverman's method and about 4 times faster than conventional correlation for the test problems.

4.2 Fast Correlation Coefficient

The purpose of this section is to compute the upper bound of the correlation coefficient R, in terms of multiplications and additions and to propose faster methods for computing R for two dimensional data, i.e., picture and window matching. The notion of running sums will be used to save computations. In picture matching, for finding the best template match, a similarity measure like normalized cross correlation is often used, which is given by:

$$R = \frac{\text{Sum}\{(x-\text{xmean})*(y-\text{ymean})\}}{\text{Sqrt}\{(\text{Sum}(x-\text{xmean})**2)*(\text{Sum}(y-\text{ymean})**2)\}},$$

where x,y are the n-tuple data vectors, and xmean,ymean are the respective means. For computational efficiency, this can be written as:

$$R = \frac{n*\text{Sum } x*y - \text{Sum } x*\text{Sum } y}{\text{Sqrt}[\{n*\text{Sum } x**2-(\text{Sum } x)**2\}*\{n*\text{Sum } y**2-(\text{Sum } y)**2\}]}.$$

Consider a large picture X of size L*L and a small window Y of size M*M as shown in Fig. 51. The total number of reference points where this window can be displaced is $(L-M+1)^2$. The number of multiplications and additions for a conventional method for computing Sum x and Sum x^2 terms are $(L-M+1)^2*M^2$ and $(L-M+1)^2*(M^2-1)$ respectively. It is shown that by using algorithm F, the computations can be reduced to $L*(2L-M)$ and $(2L-M)^2-1$ multiplications and additions

respectively. Also, algorithm F's functions are monotonic decreasing with respect to M, as M increases. As M-->L, the functions are L^2 and L^2-1, where as M-->0, the functions are $2L^2$ and $4L^2-1$.

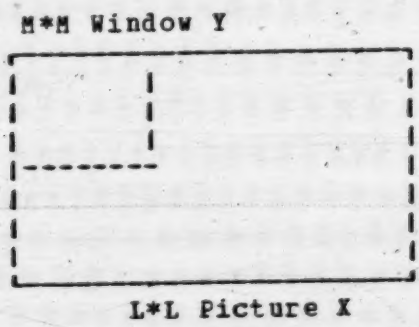

Fig. 51. Large and small pictures.

It should be noted that since the terms Sum y, Sum y^2 in R remain the same, they have to be computed only once. Also, the upper bound will only be considered for terms like Sum x, Sum y, Sum x*y, Sum x^2 and Sum y^2. The total number of operations for Sum x^2, Sum y^2, and Sum x*y terms are M^2 multiplications and M^2-1 additions. Similarly for Sum x and Sum y terms, the total number of operations are 0 multiplications and M^2-1 additions. The total number of multiplications for the conventional method are:

$$M^2+(L-M+1)^2*(M^2+M^2).$$

The first factor consists of M^2 multiplications for the term Sum y^2, and the second factor is for M^2 multiplications for the terms Sum x*y, Sum x^2 for all $(L-M+1)^2$

reference points. The total number of additions for the conventional method are:

$$2*(M^2-1)+3*(M^2-1)*(L-M+1)^2.$$

Since Sum y, Sum y^2 terms are to be computed once only and Sum x*y may have to be computed for all points, the only place one can speed up the process is in the computation of Sum x and Sum x^2 terms. The Sum x*y term can be efficiently computed by convolution, see [4]. Therefore algorithms have been designed to compute Sum x and Sum x^2 terms efficiently.

4.2.1 Algorithm E for Sum x and Sum x^2

Suppose a window W is being matched with the large picture, as shown in Fig. 52. Let Sum a1 be the horizontal sum of the first four elements of the first row and Sum $a1^2$ be the sum of squares of first four elements of the first row. Then Sum a1, Sum a2, Sum a3, Sum a4, Sum $a1^2$, Sum $a2^2$, Sum $a3^2$, Sum $a4^2$, are computed, where the range of the row sum is from 1 to 4.

Fig. 52. Computation of running sums.

Next, for computing correlation for a displacement of

1 in the column direction, the terms Sum a5, Sum $a5^2$, are added to Sum a2, Sum $a2^2$, Sum a3, Sum $a3^2$, Sum a4, Sum $a4^2$ respectively. Thus for every column displacement, only one row has to be computed, i.e, M multiplications and 4(M-1) additions, instead of M^2 multiplications and $2(M^2-1)$ additions. When the template is moved to the right, for displacement in the x-direction, this procedure can be repeated. The algorithm can be summarized, as follows:

Step 1: Compute a1,a2,a3,...,aM for an M*M window.

Step 2: Compute aM+1,aM+2,...,aL which gives the ith column of the correlation matrix.

Step 3: Compute a1,a2,a3,...,aM for the (i+1)th column.

Step 4: Repeat steps 2-3 for all possible displacements.

The total number of multiplications for Algorithm E are:

$$M^2+(L-M+1)^2*M^2+Multa,$$

where

$$Multa = (L-M+1)*(M^2+M*(L-M)).$$

The first and second factors are contributions from the terms Sum y^2 and Sum x*y respectively. The first term in Multa is the contribution from the initial processing, the second factor has M multiplications per L-M rows per (L-M+1) points. The term Multa can be written as:

$$L*M*(L-M+1).$$

The total number of additions are:

$$2(M^2-1)+(L-M+1)^2*(M^2-1)+addx+addx^2,$$

where

addx = (L-M+1)*((M-1)*M+M-1)+2(L-M+1)*(M-1)*(L-M).

The first term is the contribution from Sum y and Sum y^2 and the second term is due to Sum x*y. The expression for addx can be simplified to:

$$(L-M+1)*(2M*L-M^2-2L+2M-1).$$

Similarly $addx^2$ term has also the same value.

4.2.2 Algorithm F for Sum x and Sum x^2

The motivation of this algorithm is to compute running sums in two dimensions, with no extra storage. The operations are done 'in place', with the exception of two vector arrays of maximum row size. This algorithm is an improved version of algorithm E.

Firstly, row sums are computed for the range of 1 to M. This sum is again summed, i.e., now a column sum (two dimensional sum) is obtained. Next, while the window moves down for a column displacement, only one row is summed horizontally, the previous row sum (row refers to the window which it has left upon this displacement) is subtracted from the two dimensional sum, and the new row summed is added to the two dimensional sum. In this manner, column processing is done. When window moves right, the procedure of adding one element and subtracting the previous element is followed. The original vector is modified now and processing starts again as in the column displacement. The algorithm F can be summarized as follows:

Step 1: Compute a1,a2,a3,...,aM for an M*M window, where a1 is the row sum of first row, a2 is the row sum of second row, and so on. Similarly, compute the sum of squares terms, i.e., $a1^2$, $a2^2$,...,aM^2.

Step 2: Summation of a1,a2,...,aM gives the Sum x term for the first correlation point. Save this sum as Tem. Similarly, add $a1^2$, $a2^2$,...,aM^2 and save this sum as Tem2.

Step 3: For computing the second correlation point (column), compute aM+1, add aM+1 and -a1 to the previous sum Tem. Similarly, repeat for the sum of squares term. Repeating step 3 in this way, completes the processing of first cclumn of the correlation matrix.

Step 4: When the window is moved right by a unit displacement, modify a1,a2,a3,...,aL by subtracting the previous element and by adding the next element needed. For example, for a 3*3 case,

$$a1 = a1-a11+a14,\quad a1^2 = a1^2-a11^2+a14^2,$$

where a11 and a14 are the first and fourth elements of first row. Repeat steps 3 and 4 for different columns.

Step 5: Repeat step 4 for all possible displacements.

The total number of multiplications for Algorithm P are:

$$M^2+(L-M+1)^2*M^2+Multb,$$

where

$$Multb = L*M+2(L-M)*L.$$

The first and second factors are contributions from the terms Sum y^2 and Sum $x*y$. The first factor in Multb is from M multiplications for L rows and the second factor is 2 multiplications per L rows per (L-M+1-1) columns. The total number of <u>additions</u> are:

$$2(M^2-1)+(L-M+1)^2*(M^2-1)+addx+addx^2,$$

where

$$addx = L*(M-1)+2(L-M)*L+(M-1)*(L-M+1)+2(L-M)*(L-M+1).$$

The first factor is the contribution from Sum y and Sum y^2 terms and the second factor is due to Sum $x*y$ term. The term for $addx^2$ is the same as addx. The first factor in addx is (M-1) additions per L rows (once only, column sum), the second factor is for 2 additions per column for (L-M+1-1) columns. The third factor is the (M-1) additions (first time) per column for (L-M+1) total columns, and the last factor is for 2 additions per row (L-M) for (L-M+1) total rows. The addx factor can be simplified to:

$$(2L-M)^2-1.$$

The number of <u>multiplications</u> for conventional, algorithm E, and algorithm F can be summarized as follows:

$M^2+(L-M+1)^2*M^2+(L-M+1)^2*M^2$ for conventional,

$M^2+(L-M+1)^2*M^2+(L-M+1)*L*M$ for algorithm E,

$M^2+(L-M+1)^2*M^2+L*(2L-M)$ for algorithm F.

Similarly, the number of <u>additions</u> are:

$2(M^2-1)+3(L-M+1)^2*(M^2-1)$ for conventional,

$2(M^2-1)+(L-M+1)^2*(M^2-1)+2(L-M+1)*(2L*M-M^2-2L+2M-1)$ for algorithm E,

$2(M^2-1)+(L-M+1)^2*(M^2-1)+2(2L-M)^2-2$ for algorithm F.

The number of multiplications and additions are given in the first and second rows of Table XV. The ratio of the operations taken by the conventional method as compared to the algorithm F is also given. The ratio indicates that algorithm F is about 1.9 to 2 times faster and about 2.6 to 3 times faster than the conventional method in the number of multiplications and additions, respectively. It should be noted that L is usually very large with respect to M.

Table XV. Number of Multiplications and Additions for Matching Methods.

L	M	CONVENTIONAL	ALG. E	ALG. F	RATIO
256	8	0.7936E+07	0.4478E+07	0.4097E+07	1.937
		0.1172E+08	0.5667E+07	0.4414E+07	2.655
256	16	0.2974E+08	0.1586E+08	0.1500E+08	1.983
		0.4443E+08	0.1840E+08	0.1530E+08	2.903
256	64	0.3051E+09	0.1557E+09	0.1527E+09	1.998
		0.4576E+09	0.1635E+09	0.1529E+09	2.992
256	128	0.5453E+09	0.2769E+09	0.2728E+09	1.999
		0.8179E+09	0.2853E+09	0.2730E+09	2.997
256	192	0.3115E+09	0.1590E+09	0.1559E+09	1.999
		0.4673E+09	0.1638E+09	0.1560E+09	2.995
256	256	0.1966E+06	0.1966E+06	0.1966E+06	1.000
		0.3277E+06	0.3277E+06	0.3277E+06	1.000
10000	10	0.1996E+11	0.1098E+11	0.1018E+11	1.961
		0.2965E+11	0.1348E+11	0.1068E+11	2.776
10000	100	0.1961E+13	0.9902E+12	0.9805E+12	2.000
		0.2941E+13	0.1019E+13	0.9810E+12	2.998
10000	1000	0.1620E+15	0.8111E+14	0.8102E+14	2.000
		0.2431E+15	0.8136E+14	0.8102E+14	3.000
10000	10000	0.3000E+09	0.3000E+09	0.3000E+09	1.000
		0.5000E+09	0.5000E+09	0.5000E+09	1.000

4.3 Discussion of Results

In this chapter, template matching methods are discussed. The need for using a normalized correlation coefficient as a similarity measure is pointed out, and a fast template matching method is developed. For one test problem, Barnea and Silverman's method did not locate the correct match point. The proposed method is about 2 times faster than Barnea and Silverman's method and about 4 times faster than conventional correlation for the test problems. The upper bound for the correlation coefficient is computed in terms of the number of multiplications and additions. Two algorithms for fast correlation are given. The number of multiplications for picture matching can be reduced from

$$M^2+2(L-M+1)^2*M^2 \text{ to}$$

$$M^2+(L-M+1)^2*M^2+L*(2L-M).$$

The number of additions can be reduced from

$$2(M^2-1)+3(L-M+1)^2*(M^2-1) \text{ to}$$

$$2(M^2-1)+(L-M+1)^2*(M^2-1)+2(2L-M)^2-2.$$

Algorithm F is about 1.9 to 2 times faster than the conventional method in the number of multiplications, and is about 2.6 to 3 times faster in the number of additions.

Chapter V

CHANGE DETECTION

Changes in a scene can result from both relative movement of objects and modifications in stationary objects. To detect a motion change, it is usually assumed that objects change in displacement only. If this restriction is removed, and changes in displacement, rotation and scaling are permitted, detection of motion changes by a machine becomes cumbersome. Changes in stationary objects occur due to shape, intensity, texture, etc., or some combination of them. Each factor introduces a type of change, which a general change detection system may not be able to detect. This chapter will deal with changes due to intensity, shape and location of objects in various scenes. An application in machine vision is that of a walking robot searching for different things. Various other applications of change detection include crop growth, detection of forest fires and natural diasters, environmental changes, military reconnaissance, land use management, urban growth, etc.

In the following paragraphs, various techniques of change detection will be surveyed, and the changes will be classified into different types. The technique developed for detecting changes is described and the results obtained

on applying them on five test sets of pictures are discussed in the section 5.3.

5.1 Survey

Kawamura [29] describes an automatic change detection system for city planners. The two pictures are registered by least squares analysis, and then divided into small regions called cells. Measurements like correlation, entropy and probability are calculated for each cell pair. Using pattern classification techniques, the cells are classified into change category and no-change category. This system was the first automatic system for change detection.

Quam [45] has designed a system for detecting differences between pictures of the planet Mars. The two pictures are registered by a variant of the control point method, see Section 3.2. For corresponding sub-pictures, a two dimensional cross correlation matrix is computed for the integer shift parameters. By using interpolation between the correlation values, shift parameters in terms of fractional accuracy are obtained. The registered pictures are then subtracted pixel by pixel and displayed.

Lillistrand [35] describes a system, where portions of a scene are processed one at a time. Each sub-picture is cross correlated to obtain the shift parameters and then the sub-pictures are geometrically corrected to remove distortions. After applying the transformation, the registered

sub-picture is generated and changes between the sub-pictures are detected. The system has the further capability of suppressing shadows or other features which are not of interest.

Ulstad [61] reports an algorithm for detecting small differences between two pictures. It consists of applying nonlinear spatial registration, then matching central moments of local areas for normalization of data, and finally, computing the pixel by pixel difference picture. The point of matching the local mean and variance is emphasized for accurate computation of the difference picture.

Price and Reddy [44] approach the change detection problem differently. The two pictures are registered by the control point technique and are then segmented into regions by thresholding the histograms. The statistical features of the regions are computed and then compared for any type of change to be detected. This is one of the first symbolic approaches to the change detection problem.

For land use changes, much work has been done at the U.S. Geological Survey [38,66]; however, the system is not automatic. Changes are identified by image interpreters who mount the two pictures on a zoom transfer scope. Land which is in the process of development or under construction can be recognized, by comparison with previous photographs. An automatic digital system would be very useful for this application.

Welch and Pannell [71] have used digital pictures for land use changes in China. Pattern classification algorithms are used for classifying individual pixels into various classes. The classification maps are compared for urban growth and land use in various sectors. The authors discuss the significance of changes for land use.

Ellefsen [16] begins by geometrically registering and correcting the two LANDSAT pictures. The four bands of both pictures are used to generate a new picture, such that every pixel has 8 gray levels assigned to it. The pixels which are similar to each other in the new picture, are grouped. A spectral map of various groups is obtained for interpretation. The process is iterative in the sense that, if any group cannot be identified properly, re-grouping of pixels is done again.

De Gloria et al [14] also use LANDSAT pictures to detect spring and summer changes. They do not mention any registration steps for picture alignment. The changes are obtained by manual as well as by digital means. The input pictures are classified into various categories such as water, snow, meadow, trees, etc., by applying pattern classification algorithms. The number of pixels belonging to each category is counted for each picture. The change in the number of pixels from one date to another is then computed. A similar approach has been followed by Christenson and Lachowski [11].

5.2 Type of Changes

A change can take place either in a short time or in a long time. A rapid change takes less time to occur, e.g., forest fires, oil leakage in sea, landslides, etc. A slow change takes long time to develop, e.g., land use pattern of developing cities, cutting and planting of forests, plant growth, etc. The type of changes can be classified as: small, large, false, pre-specified, motion, intensity, and texture. Each type of change is heuristically summarized as:

1. small: A small change is hard to see visually and is minute in size. An algorithm should be able to detect this change, as it may be important for military applications.
2. large: A large change is easy to locate, both visually and algorithmically.
3. false: A false change is important to detect, since it may occur due to many reasons, e.g., incorrect results produced by an algorithm. A change may exist, but an algorithm interprets it as a no-change and vice versa. It can also result from difference pictures which have been subtracted from unnormalized pictures.
4. pre-specified: In a pre-specified change, it is already known that a change has occurred at some location or in some sub-region, e.g., monitoring of the missile sites.
5. motion: In a motion change, some objects may have moved with respect to the other objects in the same scene,

e.g., monitoring the number of cars at a traffic circle by a robot or a TV camera.

6. intensity: An intensity change is due to variation in gray levels, because of different environmental conditions.
7. texture: The texture of objects can change with respect to time, e.g., plants have different texture at different stages of growth.

5.3 Method of Solution

In this chapter, the change detection problem will be studied with respect to the computation of difference pictures. If pictures have been obtained under identical conditions, then difference picture analysis will show changes clearly; however, under different conditions, normalization of pictures is essential. It should be noted that even though the pictures may be normalized, a registration error of more than 2 pixels will produce an incorrect difference picture, i.e., changes may not be reliable. The difference picture consists of the absolute difference of the gray levels of the two pictures. In order to know the magnitude of the gray level, which can be classified as a change, the threshold T should be computed. The selection of this parameter is not easy and is dependent on the nature of changes to be detected. If the range of the difference picture is (a,b), then gray levels from a to

T are assigned a gray level of 0 and from T+1 to b are assigned a gray level 1. The resulting picture is a binary change picture, where a gray level of 0 represents a change pixel and a gray level of 1 represents a no-change pixel.

If the binary picture contains noise or areas where registration accuracy is in question, an N*N averaging can be done for smoothing purposes. Isolated change pixels (0's) can be removed by an N*N cleaning operation, i.e., in a neighborhood size of N*N, the number of pixels with gray level 0 is found. If this number is less than (N*N)/2, then the gray level of the pixel is changed from 0 to 1.

For registered pictures, the absolute difference between the gray levels of the corresponding pixels will vary. Also, for a particular gray level, the difference does not remain the same, as the spatial coordinates of the pixels change. For example, the first picture may have many pixels with a gray level of 5. To the corresponding pixels of the second picture, the gray levels may vary from 1 to 30. A one dimensional histogram for gray level 5 can then be computed, which would give the frequency of numbers in the range (1,30).

The notion of a one dimensional histogram can now be easily extended tc <u>two dimensional histograms</u>, by considering other gray levels in the first picture. The two dimensional histogram can be represented by a matrix, where the rows of the matrix are the frequencies of gray levels in the second picture, corresponding to the first picture's

gray level. The two dimensional histogram can be defined as:

Definition 6: Let a1,a2,...,aN and b1,b2,...,bN be the gray levels of the first and second picture. Let Hi1,Hi2,...,HiN be the frequency of the mapping of the corresponding pixels for gray level ai to the gray levels b1,b2,...,bN. The matrix is given by:

	b1	b2	b3	...	bN
a1	H11	H12	H13	...	H1N
a2	H21	H22	H23	...	H2N
..	...	...	...	...	...
aN	HN1	HN2	HN3	...	HNN

Given two registered pictures and a binary change picture, the notion of the two dimensional histogram can be used to generate the change and no-change histograms, by considering only those pixels which have changed or not changed. Since the mapping for a particular gray level of the first picture to the set of gray levels of the second picture is one to many, the weighted average of the gray levels of the set is computed. This average is the mapping of the particular gray level in the first picture to the second picture. The mean deviation is then the absolute difference of the two gray levels. The procedure for computing the two dimensional histogram is summarized in algorithm G.

Algorithm G:

Step 1: Given original, registered and binary change pictures, generate change and no-change two dimensional histograms according to the following criterion. Scan the three pictures pixel by pixel and repeat the following for all the pixels. For the original picture's gray level, check the gray level in the registered picture and increment the corresponding column by one. If the pixel has changed according to the binary picture, the column of the change histogram is updated, otherwise increment the no-change histogram's column.

Step 2: Compute the sum and weighted average of the numbers in a row and assign this average to the average vector. Repeat step 2 for all the rows and both the change and no-change histograms.

Step 3: Compute the deviation vectors for both the histograms between the row index and the average vector, by considering only those gray levels which are present in both the pictures. Finally, compute the means of the deviation vectors, which gives the mean deviation for change and mean deviation for no-change pixels.

The computation of the change and no-change histograms can be illustrated with the help of an example. Consider the sub-picture matrices A, B and the binary change picture

C (for a threshold of 4) as given in Tables XVI-XIX.

Table XVI. Original Matrix A.

5	5	6	6
10	10	10	10
5	5	6	6
6	6	6	6

Table XVII. Registered Matrix B.

8	7	7	7
8	8	20	20
8	8	8	7
7	7	7	7

Table XVIII. Binary Change Matrix C.

1	1	1	1
1	1	0	0
1	1	1	1
1	1	1	1

Table XIX. Two Dimensional Histograms.

	7	8	20	ave.	dev.
5	1	3	0	8	3
6	7	1	0	7	1
10	0	2	0	8	2

No-Change

	7	8	20	ave.	dev.
5	0	0	0	-	-
6	0	0	0	-	-
10	0	0	2	20	10

Change

In order to detect various kinds of change, five sets of pictures were registered, see Chapter III. The types of changes consisted of: large, slow, small, and small with

texture. In another set of pictures, the registered pictures were subtracted pixel by pixel for registration accuracy. It should be pointed out that for subtracting the two pictures, the following operation will be used:

$$p = 255 - abs(f - g),$$

where f,g are the gray levels of the two pictures and p is the gray level of the difference picture. The operation simply reverses the gray scale for displaying changes as black on the white background. The range of the Gaussian normalization for all the pictures was (0,255).

In the first test case, the two registered pictures as shown in Fig's 33 and 35, were Gaussian normalized. The resulting pictures, as shown in Fig's 53 and 54, were then subtracted pixel by pixel to generate a difference picture. The histogram of the difference picture is given in Fig. 55. From the histogram of the difference picture, the threshold was found to be 230. The difference picture was then thresholded at 230 to give a binary change picture, as shown in Fig. 56. Visually, the changes in the lake are clearly evident, which are due to snow on the lake. Empirically, the proposed algorithm was applied on the two registered pictures and the binary picture to yield a set of statistical parameters. The mean change deviation and mean no-change deviation from the first picture to the registered picture is given in Table XX (picture name Lake).

Fig. 53. Lake I with Gaussian histogram.

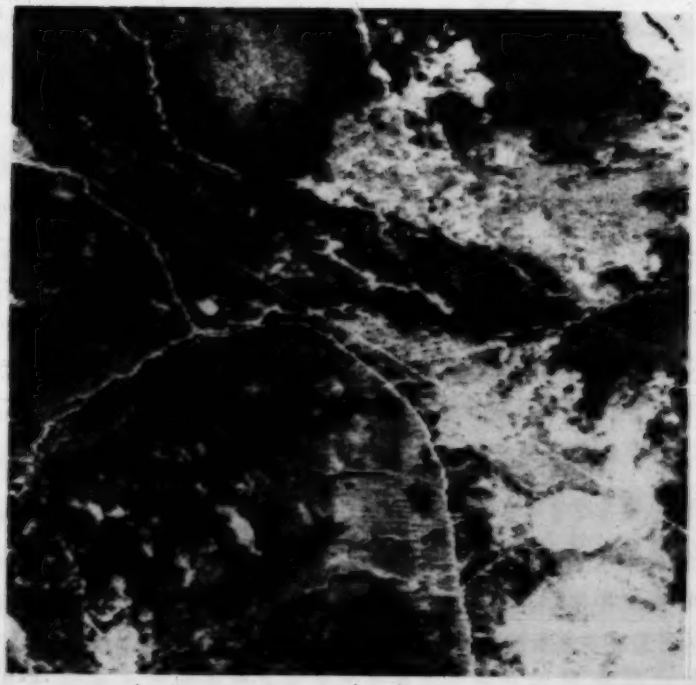

Fig. 54. Lake II (Gaussian) registered with Lake I (Gaussian) using nearest neighbor approximation.

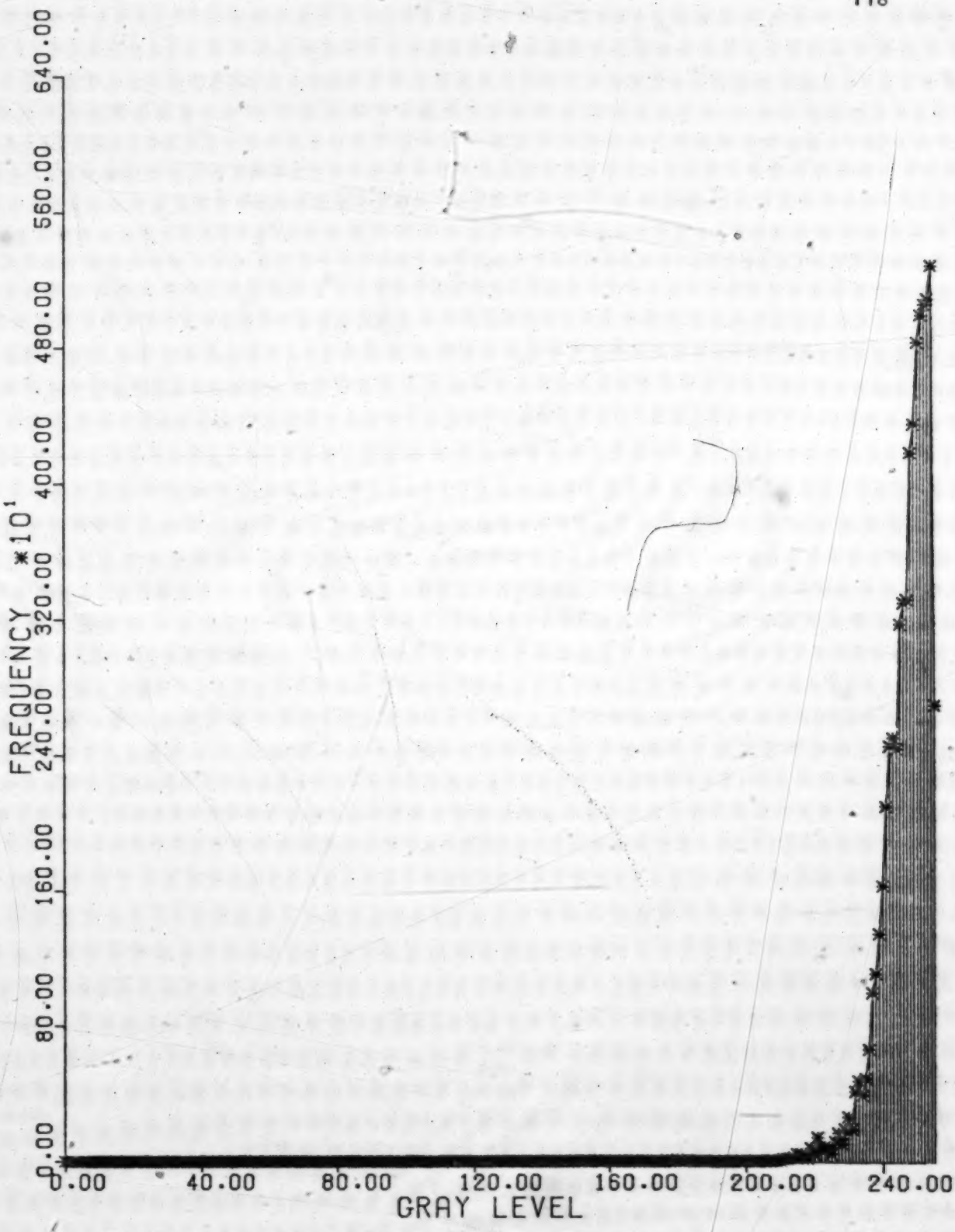

Fig. 55. Histogram of difference picture of Lake area.

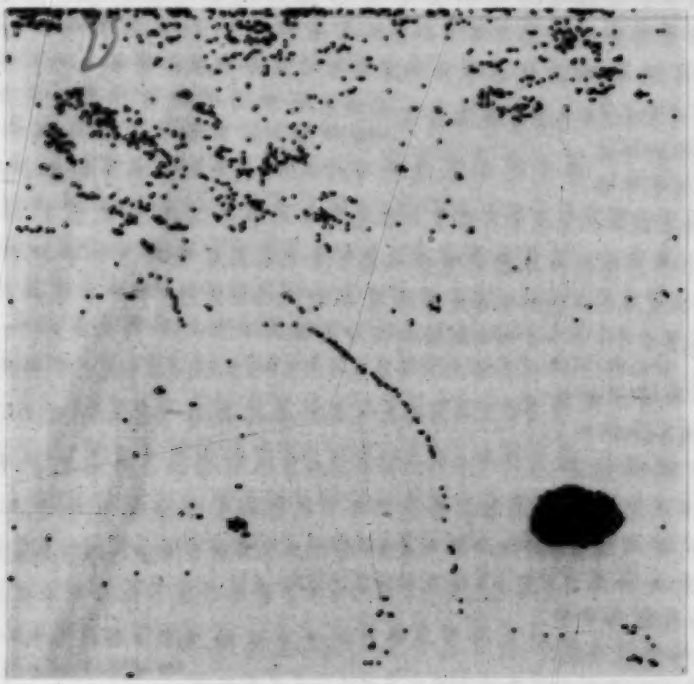

Fig. 56. Binary change picture of Lake area.

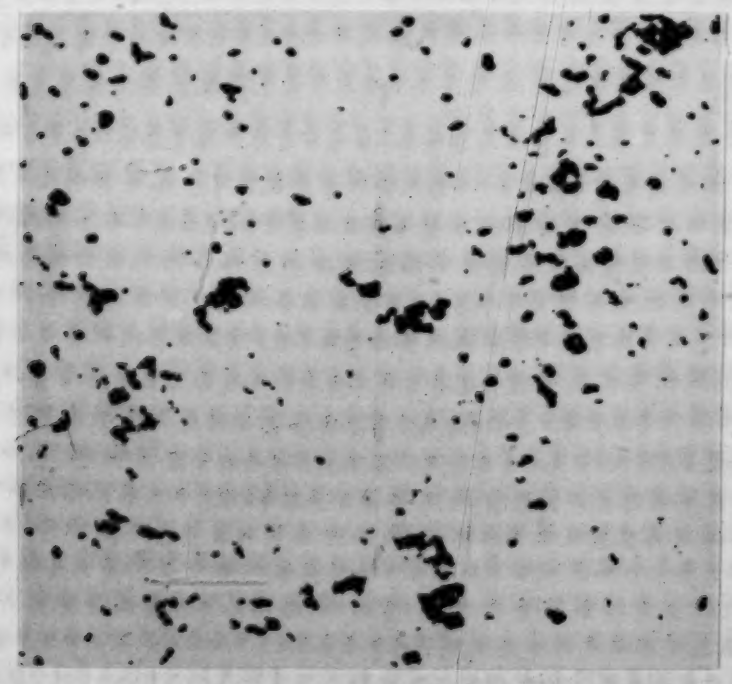

Fig. 57. Binary change picture of Edmonton using bands 4 and 5.

The number of pixels which changed is also given.

Table XX. Means of Change and No-Change Gray Levels.

Name	No-change Ave	Change Ave	Pixels Not Changed	Pixels Changed	Pixels Total
1. Lake	10.51	28.45	63567	1969	65536
2. Edmonton					
Band4	10.21	20.56	240064	10937	251001
Band5	19.87	33.59	240064	10937	251001
Band6	25.18	56.02	238072	12929	251001
Band7	22.58	68.38	238072	12929	251001
3. Aerial	6.17	21.39	81239	31161	112400
4. Cut	6.94	20.8	122231	3209	125440
5. Fertilizer					
N. N.	5.13	20.92	129548	17452	147000
Bilinear	5.06	21.64	130736	16264	147000
Cubic	4.75	23.60	131582	15417	147000

In the second test case, all four bands of the two pictures were registered. The band 5 pictures are shown in Fig's 29 and 39. The four bands of the first registered picture were then subtracted pixel by pixel, from the second registered picture to yield four difference pictures. The histograms of the difference pictures gave a threshold of T=15 for bands 4 and 5 and a threshold of T=53 for bands 6 and 7. The difference pictures were then thresholded to

give binary pictures.

Next, another binary picture was generated pixel by pixel by a logical 'or' of bands 4 and 5, e.g., if either of the pixels of band 4 or band 5 has a gray level 1, the new picture will also have a gray level 1. In the similar manner, another binary picture was generated by the logical 'or' of pixels of bands 6 and 7. This processing was done, because bands 4 and 5 and bands 6 and 7 are very close together, both visually as well as statistically. Although, some features are differentiated by each band separately, in this study this pcint will not be considered.

The two new binary pictures were then cleaned by using a 7*7 operator. The resulting pictures are shown in Fig's 57 and 58. In order to point out the land use changes, the pictures were overlayed with the registered picture and are shown in Fig's 59 and 60. Visually, the urban growth in the Edmonton area can be seen easily. The new developing areas of Castle Downs, Mill Woods, St. Albert, Carleton Estate, Albert Park, Thorncliffe, Patricia Hts., are pointed out in Fig. 59. Rural changes such as cutting of crops, etc., are evident in Fig. 60. It should be noted that bands 4 and 5 emphasize land use changes and bands 6 and 7 emphasize forestry and rural type of changes. Empirically, the mean change deviation and mean no-change deviation in gray levels for all bands is given in Table XX (picture name Edmonton). It should be noted that the cleaned binary picture (of bands 4 and 5) was used as a mask for both bands 4 and 5.

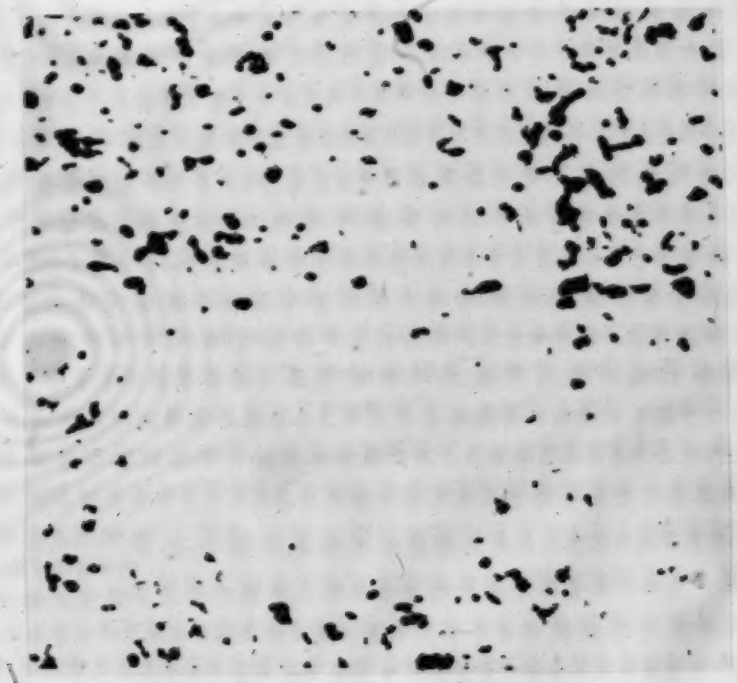

Fig. 58. Binary change picture of Edmonton using bands 6 and 7.

Fig. 59. Overlaid change picture of Edmonton II using bands 4 and 5.

Fig. 60. Overlaid change picture of Edmonton II using bands 6 and 7.

Fig. 61. Window of Edmonton II with Gaussian histogram.

Similarly, the other cleaned binary picture was used for both bands 6 and 7.

In the third test case, the registered pictures as shown in Fig's 45 and 50, were Gaussian normalized. The purpose of this test was to check the registration accuracy of pictures. The resulting pictures as shown in Fig's 61 and 62, were then subtracted pixel by pixel. From the histogram of the difference picture, the threshold was chosen to be 224. The binary change picture was obtained by thresholding the difference picture at 224. The binary change picture as shown in Fig. 63, show large changes in four years. The registration accuracy was found to be reasonable, i.e., within 1 to 2 pixels. The mean change deviation and mean no-change deviation in gray levels is given in Table XX (picture name Aerial).

In the fourth test case, the registered pictures as shown in Fig's 42 and 41, were Gaussian normalized. The resulting pictures as shown in Fig's 64 and 65, were then subtracted pixel by pixel to generate a difference picture. From the histogram of the difference picture, the threshold was chosen to be 234. The difference picture was then thresholded at 234 to give a binary change picture. The binary picture was cleaned using a 3*3 operator to yield the final binary change picture, as shown in Fig. 66. Visually, the small changes such as cutting of forests were detected easily. It should be pointed out that cleaning does

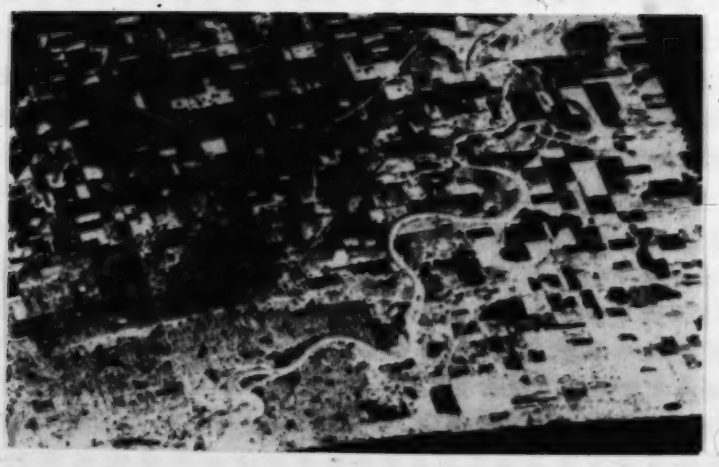

Fig. 62. Aerial picture (Gaussian) registered with Edmonton II (Gaussian) using proposed approximation.

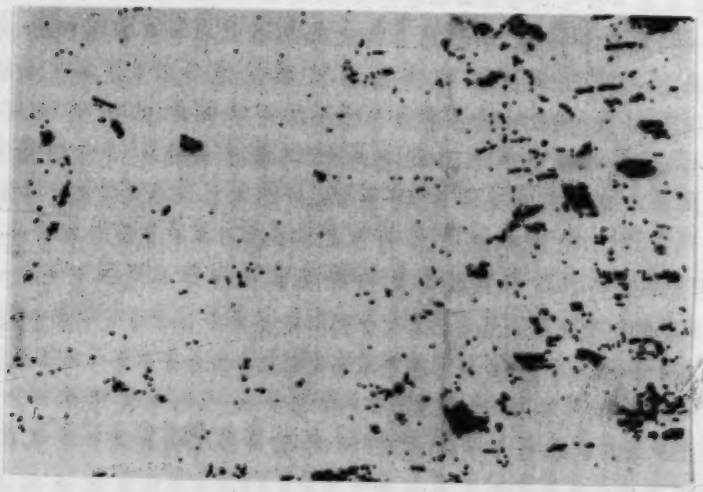

Fig. 63. Binary change picture of aerial and Edmonton II.

Fig. 64. Forestry Cut I (Gaussian) registered with Forestry Cut II (Gaussian) using nearest neighbor approximation.

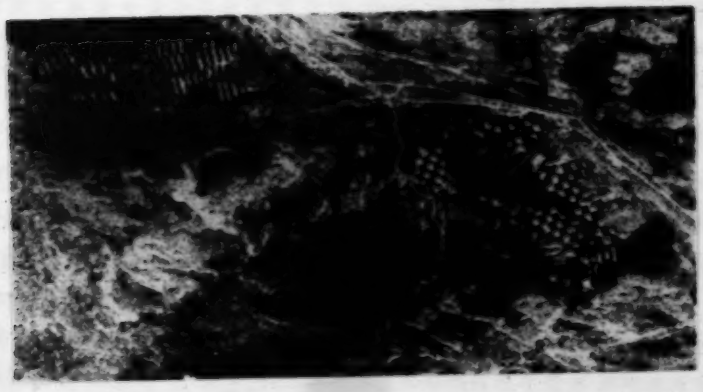

Fig. 65. Forestry Cut II with Gaussian histogram.

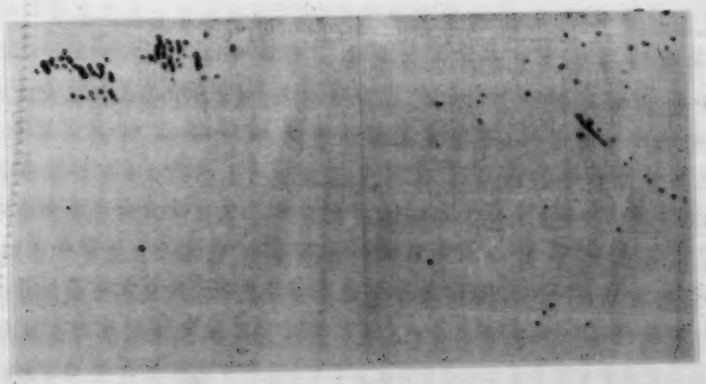

Fig. 66. Binary change picture of cut area.

in fact alter the shape of the detected changes. The mean change deviation and mean no-change deviation in gray levels is given in Table XX (picture name Cut).

In the final test case, the Fertilizer 2 picture and the registered Fertilizer 3 pictures, as shown in Fig's 31, 36-38, were Gaussian normalized. The resulting pictures are shown in Fig's 67-70. The normalized Fertilizer 2 picture was then subtracted from the three normalized registered pictures to give three difference pictures. From the histograms of the difference pictures, the threshold was chosen to be 237. The difference pictures were then thresholded at 237 to give three binary pictures. These pictures were cleaned using a 3*3 operator to give the final change pictures, as shown in Fig's 71-73.

Visually, the small changes detected are good. Since the pictures have texture, the difference pictures and subsequently the binary pictures, show spikes of noise. It should be noted that texture was not taken into account for detecting changes. Thus this test case raises the question of using texture information in detecting changes for texture oriented pictures. Also, the change picture obtained through the cubic convolution interpolation is much better than that obtained by the nearest neighbor approximation. The mean change deviation and mean no-change deviation in gray levels for both the nearest neighbor and bilinear approximation is given in Table XX (picture name Fertilizer).

Fig. 67. Fertilizer 2 with Gaussian histogram.

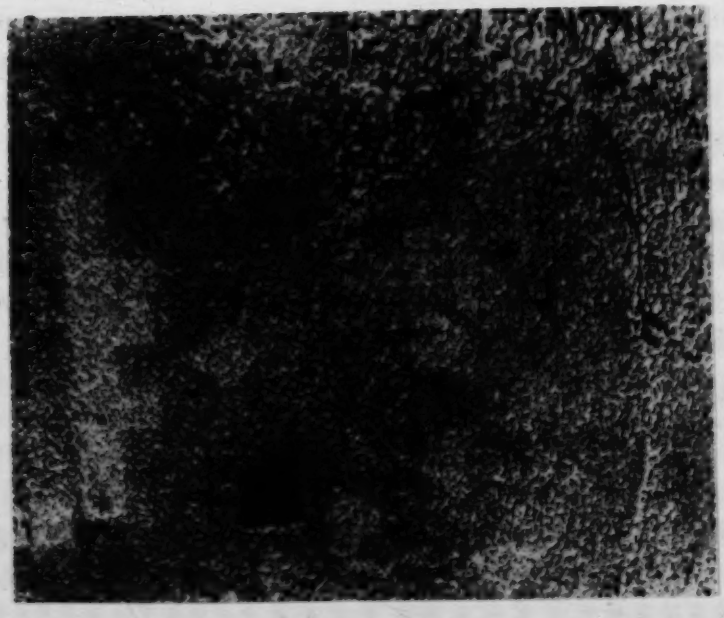

Fig. 68. Fertilizer 3 (Gaussian) registered with Fertilizer 2 (Gaussian) using nearest neighbor approximation.

Fig. 69. Fertilizer 3 (Gaussian) registered with Fertilizer 2 (Gaussian) using bilinear approximation.

Fig. 70. Fertilizer 3 (Gaussian) registered with Fertilizer 2 (Gaussian) using cubic convolution.

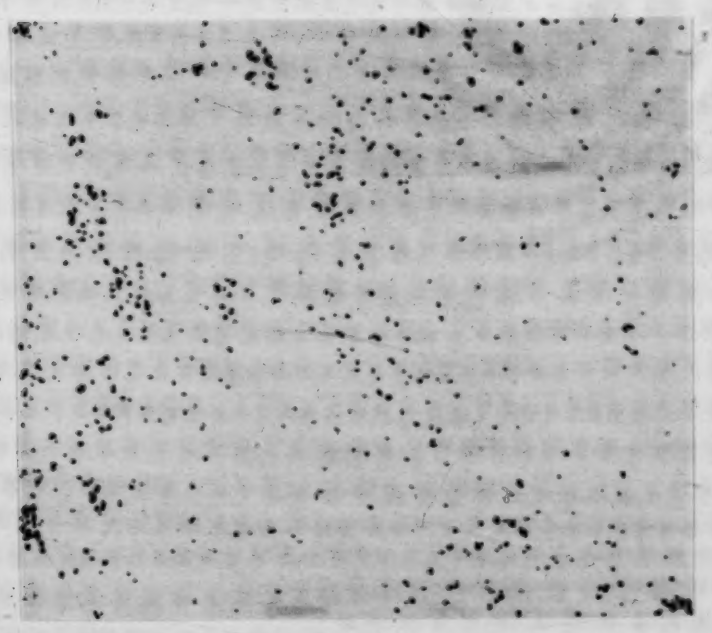

Fig. 71. Binary change picture of Fertilizer area using nearest neighbor approximation.

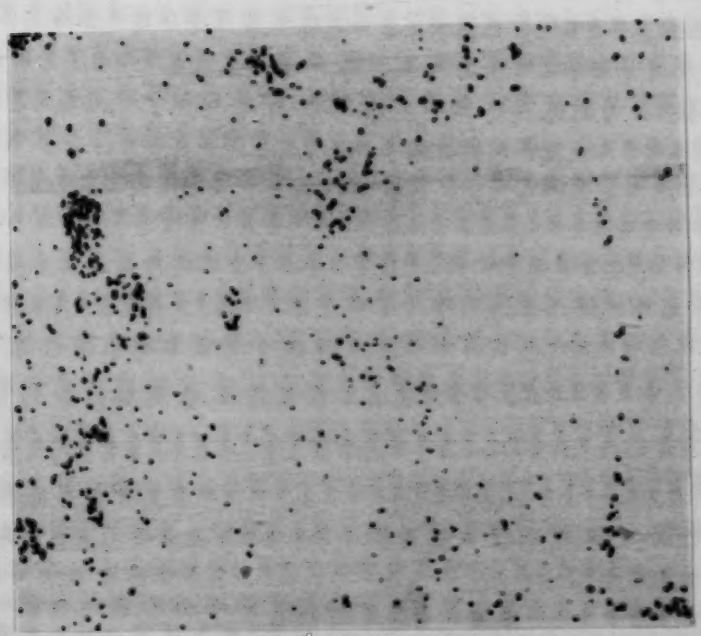

Fig. 72. Binary change picture of Fertilizer area using bilinear approximation.

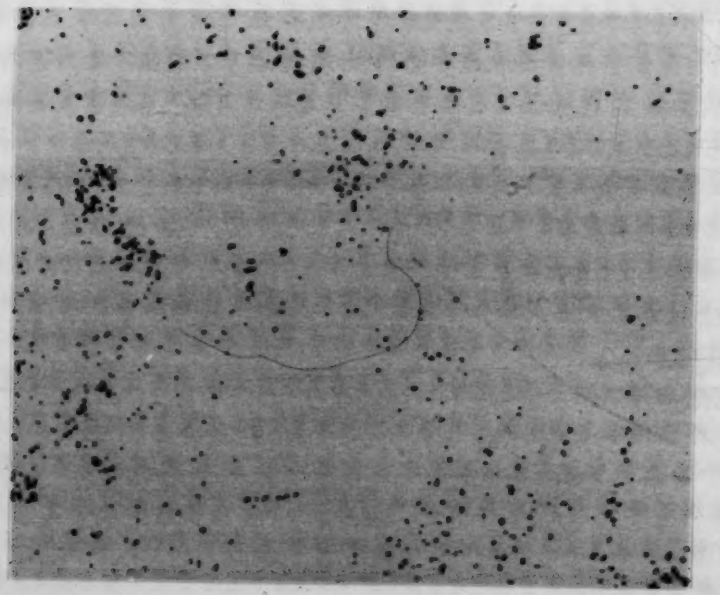

Fig. 73. Binary change picture of Fertilizer area using cubic convolution.

5.4 Discussion

This chapter is concerned with detecting various types of changes in scenes. The changes are classified into different categories. An algorithm for a two dimensional histogram is given for computing the deviation in gray levels from one registered picture to the another. Five sets of registered pictures are considered for identifying various kinds of changes. The pictures are Gaussian normalized and subtracted pixel by pixel. The difference pictures are then thresholded to yield binary change pictures. Isolated change pixels are removed by a cleaning operation. The changes identified are the surface of a lake during fall and winter seasons, forestry cut area, fertilizer area, and rural and urban land use. The mean change deviation for all the sets of pictures was found to be higher than the mean no-change deviation.

CHAPTER VI

CONCLUSIONS

The aim of this research was to study the problems of producing an automatic change detection system. This involved normalization, automatic control point selection, registration, template matching and detection of changes.

The picture normalization method has the advantage that it can map the original gray scale [a,b] to any desired range [c,d]. Normalization enhances and generates the desired standard pictures.

In normalization, averaging over the diamond neighborhood was done to compute the average gray level. The neighborhood size can be varied and local information can be used to obtain better normalized pictures.

The picture enhancement method sharpens pictures and the binary gradient pictures of the enhanced pictures have less noise and thin contour lines. The enhancement method was not used directly in detecting changes; however, it can be used in automatic control point selection. The original pictures can first be enhanced and then control points can be selected from the enhanced pictures.

In enhancement, the neighborhood size for defining binary relations, can be varied. The method can also be

used for reducing the number of gray levels in the pictures to some desired number.

In automatic control point selection, control points were selected on the edges of the objects of the pictures. The problems encountered in their selection are presented and some basic features of control points are given.

If a measure for a good control point is defined, the measure may vary from picture to picture. Fixed templates or pre-specified objects such as lakes do not offer a general solution. The connected edge elements approach has been successful but some more work has to be done in this area. The pictures can also be averaged to reduce the complexity problem. The control points can then be found on reduced versions.

In the registration of pictures, the iterative feedback technique is fast and accurate, as it unifies the control point and the correlation approaches. The registration accuracy for the first order transformation was about 1 to 2 pixels and for the second order transformation was about 1 pixel. The cubic convolution technique of computing the gray level of the output pixel was better than the nearest neighbor approximation, as the picture generated by the nearest neighbor transformation was non-smooth.

For checking registration accuracy, either the complete pictures can be correlated or small corresponding windows can be correlated at some check points. The number of check points with a confidence interval of more than 95%

may be obtained by applying techniques of statistical sampling theory. It should be pointed out that the potential control points serve as good test points for checking registration accuracy.

In registration with large scale differences, registration of an aerial picture (2048*2048) with a LANDSAT picture (600*600) has been done. The transformed picture was generated by different approaches, e.g., nearest neighbor, averaging and the proposed approximation. The transformed picture obtained by the proposed technique is visually the best, as the new picture is smooth and noise free.

The proposed technique for large scale differences can also be improved by using either bilinear or spline interpolation, instead of zeroth order approximation. To be more accurate, the contribution of the irregular area within the boundary should also be taken into account. This can best be dealt with by spline functions.

In matching templates, the proposed algorithm was about 2 times faster than Barnea and Silverman's method and about 4 times faster than the standard correlation method. The number of operations in computing the normalized correlation coefficient are evaluated and faster algorithms to reduce the number of operations are reported.

The technique developed for fast template matching can also be improved. The values of the pre-specified constants k1, k2 and k3 should either be pre-computed or obtained

through some measures. Also, faster algorithms for computing the correlation coefficient's term Sum x*y, should be developed.

In detecting changes between two registered pictures, various types of changes were detected. In the change detection procedure, stress has been laid on the binary change picture and the computation of two dimensional histograms. Because of fixed thresholding in the difference pictures, some information is lost. Techniques should be developed to filter out the information, which is indeed needed. Specific applications of the binary change detection technique include, urban development using visible wavebands (bands 4 and 5 of LANDSAT), forest fire detection systems where fires produce much stronger signals in the thermal infrared region of the spectrum and snow and ice surveys on rivers and lakes in areas where there is no urban development.

Also, due to cleaning and smoothing, some change pixels are lost. Even though, some isolated pixels may be valid change pixels, cleaning would change them to be no-change pixels. In this thesis, isolated pixels have not been considered to be valid. If due to cleaning, change pixels comprising the objects disappear, better techniques should be used to take care of this problem. Texture information was not used in this change detection study. The fertilizer pictures are texture oriented, which cause the binary change pictures to have many random noise points.

The role of texture in discriminating various types of changes should be studied.

REFERENCES

1. H.C. Andrews, Computer Techniques in Image Processing, Academic Press, New York, 1970.
2. H.C. Andrews and C.L. Patterson, Singular Value Decompositions and Digital Image Processing, IEEE Trans. Acous. Sp. and Signal Processing, Vol. ASSP-24, 1976, pp. 26-53.
3. H.C. Andrews, Digital Image Restoration: A Survey, Computer, Vol. 7, May 1974, pp. 36-45.
4. P.E. Anuta, Spatial Registration of Multispectral and Multitemporal Digital Images using FFT techniques, IEEE Trans. on Geoscience Electronics, Vol. GE-8, 1970, pp. 353-368.
5. P.E. Anuta, Digital Registration of Multispectral Video Imagery, SPIE Journal, Vol. 7, 1969, pp. 108-175.
6. D.I. Barnea and H.F. Silverman, A class of Algorithms for Fast Digital Image Registration, IEEE Trans. Computers, Vol. C-21, 1972, pp. 179-186.
7. Bendix Aerospace System Divison, Multispectral Picture Registration Study, Final Rept., BSR-2855, NASA Cont. NAS5-11699, 1969, 159 pp.
8. R. Bernstein, Digital Image Processing of Earth

Observation Sensor Data, IBM Journal of Res. and Development, Vol. 20, Jan. 1976, pp. 40-57.

9. R. Bernstein and D.G. Ferneyhough, Digital Image Processing System, Photogrammetric Engineering and Remote Sensing, Vol. 41, 1975, pp. 1965.

10. F.C. Billingsley, Review of Digital Image Processing, EUROCOMP 75, London, Sept. 1975.

11. J.W. Christenson and H.M. Lachowski, Urban area delination and detection of Change along the urban-rural boundary as derived from LANDSAT digital data, ACSM-ASP Conference, 1976.

12. W.A. Davis, R.N. McPherson, M.W. Smith and N. Tsang, A System for processing of Digital Pictures, Proc. 1973 NRC Man-Computer Communications Conf., Ottawa, Canada, 1973.

13. W.F. Derouchie and R.B. Forrest, Potential positioning accuracy of ERTS1 MSS Imageing, ACSM-ASP Conference, Missouri, 1974.

14. S.D. De Gloria, S.J. Daus, N. Tosta and K. Bonner, First Linutilization of high altitude photography and LANDSAT-1 data for Change Detection and Sensitive area Analysis, 11th Int. Symp. on Remote Sensing of Environment, Michigan, 1975.

15. R. Eberlein, G.J. Vanderburg, A. Rosenfeld and L.S. Davis, Edge and Line Detection in the ERTS Imagery: a comparative study, Rept. TR-312, University of Maryland, June 1974.

16. R. Ellefson, Automatic Digital Processing of ERTS-1 MSS data in an Urban Land Use Mapping experiment, American Society of Photogrammerty meeting, St. Louis, Missouri, March 1974.

17. W. Feller, An introduction to Probability theory and its applications, Vol. I, John Wiley and Sons, New York, 1968.

18. E.A. Fleming, Positioning of Shore Features with the aid of LANDSAT imagery, Photogrammetric Engineering and Remote Sensing, Vol. 43, 1977, pp. 53-59.

19. A. Habibi and G.S. Robinson, A Survey of Digital Picture Coding, Computer, Vol. 7, May 1974, pp. 22-34.

20. E.L. Hall, Almost Uniform Distributions for Computer Image Enhancement, IEEE Trans. Computers, Vol. C-23, 1974, pp. 207-208.

21. D.J. Hall, R.M. Endlich, D.E. Wolf and A.E. Brain, Objective methods for Registering landmarks and determining cloud motions from satellite, IEEE Trans. Computer, Vol. C-21, 1972, pp. 768-777.

22. M.J. Hannah, Computer matching of Stereo Images, Ph.D thesis, Stanford University, 1974.

23. R.M. Haralick, K. Shanmugam and I. Dinstein, Textural features for Image classification, IEEE Trans. Sys. Man and Cybernetics, Vol. SMC-3, 1973, pp. 610-621.

24. R.M. Hoffer and the LARS staff, Techniques for Computer Aided Analysis of ERTS-1 data, useful in Geologic. Forest and Water Resource Surveys, LARS Report: 121073,

Purdue University, 1973.

25. R.M. Hummel, Histogram Modification Techniques, Computer Graphics and Image Processing, Vol. 4, 1975, pp. 209-224.

26. B.R. Hunt, The Application of Constrained Least Squares Estimation to Image Restoration by Digital Computer, IEEE Trans. Computers, Vol. C-22, 1973, pp. 805-812.

27. R.R. Jayroe, J.F. Andrus and C.W. Campbell, Digital Image Registration method based on Binary Boundary Maps, NASA Tech. Note, TN D-7607, March 1974.

28. T. Kaneko, Evaluation of LANDSAT Image Registration Accuracy, Photogrammetric Engineering and Remote Sensing, Vol. XLII, 1976, pp. 1285-1299.

29. D.G. Kawamura, Automatic Recognition of Changes in Urban Development from Aerial Photographs, IEEE Trans. Sys. Man and Cybernetics, Vol. SMC-1, 1971, pp. 230-239.

30. M.D. Kelly, Edge Detection in Pictures by Computers using Planning, in Machine Intelligence, (B. Meltzer and D. Michie, Ed's.), Vol. 6, University Press, Edinburgh, 1971, pp. 379-409.

31. S.K. Kenue and W.A. Davis, Image Processing Software Routines and User's manual, Tech. Rept., Dept. of Computing Science, Univ. of Alberta, 1977.

32. S.K. Kenue and W.A. Davis, Preprocessing for Image Processing, Canadian Society for Computational Studies of Intelligence, Proc. of First CSCSI/SCEIO National

Conf., Univ. of British Columbia, Vancouver, B.C., August 1976, pp. 31-41.

33. H.P. Kramer and J.B. Bruckner, Iterations of a non-linear Transformation for Enhancement of Digital Images, Pattern Recognition, Vol. 7, 1975, pp. 53-58.

34. R.P. Kruger, Computer Processing of Radiographic Images, Ph.D thesis, Univ. of Missouri-Columbia, June 1971.

35. R.L. Lillestrand, Techniques for Change Detection, IEEE Trans. Computers, Vol. C-21, 1972, pp. 654-659.

36. D.T. Lindgren, R.B. Simpson and W. Goldstein, Land use Change Detection in the Boston and New Haven areas, Tech. Rept.: Dept. of Geography, Dartmouth College Project in Remote Sensing, Jan. 1974.

37. J.A. Leese, C.S. Novak, and B.B. Clark, An Automated technique for obtaining cloud motion from geo-synchronous satellite data using cross correlation, Journal App. Meteorology, Vol. 10, 1971, pp. 118-132.

38. V.A. Milazzo, Some findings on the applications of ERTS and SkyLab Imagery for Metropolitan land Use Analysis, 9th Int. Symp. on Remote Sensing of Environment, Michigan, 1973.

39. R.N. Nagel and A. Rosenfeld, Ordered Search Techniques in Template Matching, Proc. IEEE, Vol. 60, 1972, pp. 242-244.

40. G. Nagy, Digital Image Processing activities in Remote Sensing for Earth Resources, Proc. IEEE, Vol. 60, 1972,

pp. 1177-1200.

41. M. Onoe and M. Saito, Automatic threshold setting for the SSDA, IEEE Trans. Computer, Vol. C-25, 1976, pp. 1052-1053.

42. P.G. Peet, A. Mack and L. Crosson, Affine Transformations from aerial photographs to Computer Compatiable Tapes, 3rd Earth Resources Technical Satellite Symp., NASA, SP-351, 1973, pp. 1719-1724.

43. F.G. Peet, The Use of Invariants in the Transformation and Registration of Images, The Canadian Surveyor, Vol. 29, 1975, pp. 501-506.

44. K. Price and R. Reddy, Change Detection in Multispectral Images, Dept. of Computer Science, Carnegie-Mellon University, Tech. Report, 1976.

45. L.H. Quam, Computer Comparison of Pictures, Ph.D thesis, Stanford University, May 1971.

46. S.S. Rifman, K.W. Simon, and R.H. Caron, Second generation digital techniques for processing LANDSAT MSS data, Earth Resources Survey Symp., Texas, June 1975.

47. S.S. Rifman, Evaluation of Digitally Corrected ERTS Imagery, Symp. on Management and Utilization of Remote Sensing Data, Sioux Falls, South Dakota, Oct. 20-Nov. 1973.

48. A. Rosenfeld, Picture Processing: 1974, Computer Graphics and Image Processing, Vol. 4, 1975, pp. 133-155.

49. A. Rosenfeld, Picture Processing: 1975, Computer Graphics and Image Processing, Vol. 5, 1976, pp. 215-237.

50. A. Rosenfeld, Picture Processing: 1976, Computer Graphics and Image Processing, Vol. 6, 1977, pp. 157-183.

51. A. Rosenfeld and A.C. Kak, Digital Picture Processing, Academic Press, New York, 1976.

52. A. Rosenfeld and J. Pfaltz, Distance Functions on Digital Pictures, Pattern Recognition, Vol. 1, 1968, pp. 33-63.

53. A. Rosenfeld, Automatic Detection of Changes in Reconnaissance Data, Conf. Proceeding of the 5th. Int. Convention on Military Electronics, 1961, pp. 492-499.

54. S. Shlein, Geometric Correction, Registration and Resampling of LANDSAT Images, Remote Sensing Science and Technology Symp., Ottawa, Canada, Feb. 21-23, 1977.

55. A.J. Scott, Automatic Registration of Imagery, Technical Report, Dept. of Geography, Carleton University, Ottawa, 1976.

56. E.A. Smith and D.R. Phillips, Automated Cloud Tracking using precisely Aligned Digital ATS pictures. IEEE Trans. Computers, Vol. C-21, 1972, pp. 715-729.

57. M.M. Sondhi, Image Restoration: The Removal of Spatially Invariant Degradation, Proc. IEEE, Vol. 60, 1972, pp. 842-853.

58. D. Steiner, Digital Geometric Picture Correction using a Piecewise Zero-order Transformation, Remote Sensing of Environment, Vol. 3, 1974, pp. 261-283.

59. J.E. Taber, Evaluation of Digitally Corrected ERTS Images, 3rd Earth Resources Tech. Satellite Symp., NASA, SP-351, 1973, pp. 1837-1843.

60. M. Tasto and V. Block, Locating Objects in Complex Scenes using a Spatial Distance Measure, Proc. of the Second Int. Joint Conf. Pattern Recognition, Copenhagen, 1974.

61. M.S. Ulstad, An Algorithm for estimating small scale differences between two Digital Images, Pattern Recognition, Vol. 5, 1973, pp. 323-333.

62. R. Welch and C.W. Pannell, LANDSAT investigations of recent Urban Land Use Changes in Northeast China, 10th Int. Symp. on Remote Sensing of Environment, Michigan, 1975.

63. J.S. Weszka, E.J. Carton, J.A. Verson, J.M. Mohr and A. Rosenfeld, Some Basic Edge Degradation and Enhancement techniques, Rept. TR-278, University of Maryland, Dec., 1973.

64. P.M. Will, R. Bakis, and M.A. Wesley, On an All Digital approach to Earth Resources Satellite Image Processing, IBM Research Report: RC 3027, New York, 1970.

65. P.A. Wintz, Transform Picture Coding, Proc. of IEEE, Vol. 60, 1972, pp. 809-872.

66. J.R. Wray, A Remote Sensing System for detecting Land

Use Change in Metropolitan areas, 7th Int. Symp. on Remote Sensing of Environment, Michigan, 1971.

APPENDIX A

LIST OF SOFTWARE PROGRAMS

The user's manual and the source listing for FORTRAN programs used in various chapters is given in [31]. A list of programs with reference to each chapter and the page number of the technical report [31] is given as:

Name	Page	Chapter
AER.TRANSRC	6	III
AVE44	8	III
AVERAGE	9	II-V
BANDBIN	10	V
BILINEAR	11	III
BINARY	13	V
BYTE.ERTS	13	II-V
CLEAN	15	V
CONSEND	16	II-V
CORR.ERTS	17	III and IV
CPDUMP2	19	III
DIFERTS	21	V
DIFF	22	V
EDHIST	23	V
ENHANCE	24	II

GRAD ..... 25 II

KRAMER ..... 26 II

MATCH ..... 27 III and IV

MASK ..... 30 III

NEWBAND8 ..... 31 III, IV and V

NEWCORR ..... 33 III

NEW.AUTO ..... 35 III

NORMAL.1 ..... 37 II and V

PRINT ..... 39 II-V

SECORDER ..... 41 III

TOPDP ..... 46 II-V

TO360 ..... 47 II-V

WESZKA ..... 48 II

WINDOW ..... 49 II-V

WIN.ERTS ..... 50 II-V

APPENDIX B

SUMMARY OF PICTURE DETAILS

This appendix gives a summary of picture sizes, how they were obtained, filters used, output medium, etc. The Spinach and Neuron pictures were obtained by digitizing two covers from the IBM Journal of Research and Development. This was done by a TV camera, the output of which was converted to digital by a video quantizer [12]. For output, the digital data was then converted into an analog image by a 35mm camera and a Tektronix 611 storage scope operating in non-store mode. Other pictures were first produced on an electrostatic printer and then photographed to reduce the scale. Histograms of the pictures were drawn by a Calcomp plotter.

The LANDSAT pictures in the form of computer compatible tapes were obtained from the Canada Centre for Remote Sensing, Ottawa. The band numbers used are given in the figure description, together with the line and pixel numbers in the following format (first row, last row, first col, last col).

The aerial picture of Edmonton was obtained from the National Air Photo Library and has the following details:

NAPL Roll no: RSA 30376 Frame 3

Date: 5 Oct. 1971

Film: Kodak aerochrome IR

Filter: Pan 520

Altitude: 38600 feet

Scale: 1:12500

The fertilizer pictures were obtained from Dr. Jim Lee of the Pacific Forest Research Centre, and are of the Shawnigan Lake Fertilzer Plots on Vancouver Island, B. C. Both pictures were then digitized on a scanning microdensitometer using red, blue and green filters by the Canada Centre for Remote Sensing. Detailed information about the pictures along with other comments are listed as follows:

	Name	Page
1.	Neuron, 256*256.	16
2.	Neuron with Gaussian histogram, 256*256.	16
5.	Neuron with flat histogram, 256*256.	19
6.	Spinach, 256*256.	28
8.	Spinach with flat histogram, 256*256.	31
9.	Spinach with Gaussian histogram, 256*256.	31
10.	Edges of Neuron, 256*256.	32
11.	Edges of Neuron with flat histogram, 256*256.	32
12.	Edges of Neuron with Gaussian histogram, 256*256.	33

PICTURE DETAILS

13. Edges of Spinach, 256*256. 33

14. Edges of Spinach with flat histogram, 256*256. 34

15. Edges of Spinach with Gaussian histogram, 256*256. 34

16. Kramer Neuron, 256*256. 36

17. Kramer Spinach, 256*256. 36

18. Weszka Neuron, 256*256. 37

19. Weszka Spinach, 256*256. 37

20. Edges of Weszka Neuron, 256*256. 38

21. Edges of Weszka Spinach, 256*256. 38

22. Enhanced Neuron, 256*256. 40

23. Enhanced Spinach, 256*256. 40

24. Edges of enhanced Neuron, 256*256. 41

25. Edges of enhanced Spinach, 256*256. 41

28. Edmonton I with control points, 600*600, band 5, July 1973, Image: 1344-18071, (1416,2015,1888,2487). Only 512*512 window is shown. 58

29. Edmonton II with control points, 600*600, band 5, July 1975, Image: 1160-17553, (1687,2286,1819,2418). Only 512*512 window is shown. 58

30. Binary gradient picture of Edmonton I, 512*512, band 4. 60

PICTURE DETAILS

31. Fertilizer 2 with control points, 600*500, red band, 1973. 62

32. Fertilizer 3 with control points, 600*500, red band, 1974. 62

33. Lake I, 256*256, band 5, Sept. 1972. Image: 1043-18341, (2030,2285,2691,2946). 67

34. Lake II, 384*384, band 5, Mar. 1974. Image: 1600-18242, (15,398,1536,1919). Only 256*256 window is shown. 67

35. Lake II registered with Lake I using nearest neighbor approximation. 256*256, band 5. 69

36. Fertilizer 3 registered with Fertilizer 2 using nearest neighbor approximation, 420*350, red band. 69

37. Fertilizer 3 registered with Fertilizer 2 using bilinear approximation, 420*350, red band. 72

38. Fertilizer 3 registered with Fertilizer 2 using cubic convolution, 420*350, red band. 72

39. Edmonton I registered with Edmonton II using nearest neighbor approximation, 501*501, band 5. 73

40. Forestry Cut I, 512*512, band 5, Sept. 1972,

PICTURE DETAILS

Image: 1043-18341, (256,767,1731,2242). 75
Only 256*490 window is shown.

41. Forestry Cut II, 512*512, band 5, Oct. 1973, 75
Image: 1439-18325, (200,711,1880,2391).
Only 256*490 window is shown.

42. Forestry Cut I registered with Forestry
Cut II using nearest neighbor approximation, 75
256*490, band 5.

45. Window of Edmonton II, 281*400, band 5. 82
(100,355,200,455).

46. 4*4 average of aerial picture, 82
512*512, red band.

47. Window of aerial picture, 83
400*256, red band.

48. Average aerial picture registered with
Edmonton II using nearest neighbor 83
approximation, 281*400, red band.

49. Aerial picture registered with Edmonton II
using nearest neighbor approximation, 84
281*400, red band.

50. Aerial picture registered with Edmonton II
using proposed approximation, 84
281*400, red band.

53. Lake I with Gaussian histogram, 256*256, 117

PICTURE DETAILS

band 5.

54. Lake II (Gaussian) registered with Lake I (Gaussian) using nearest neighbor approximation, 256*256, band 5. 117

56. Binary change picture of Lake area, 256*256. 119

57. Binary change picture of Edmonton using bands 4 and 5, 501*501. 119

58. Binary change picture of Edmonton using bands 6 and 7, 501*501. 122

59. Overlaid change picture of Edmonton II using bands 4 and 5, 501*501. 122

60. Overlaid change picture of Edmonton II using bands 6 and 7, 501*501. 123

61. Window of Edmonton II with Gaussian histogram, 281*400, band 5. 123

62. Aerial picture (Gaussian) registered with Edmonton II (Gaussian) using proposed approximation, 281*400, red band. 125

63. Binary change picture of aerial and Edmonton II, 281*400. 125

64. Forestry Cut I (Gaussian) registered with Forestry Cut II (Gaussian) using nearest neighbor approximation, 256*490, band 5. 126

PICTURE DETAILS

65. Forestry Cut II with Gaussian histogram, 256*490, band 5. 126

66. Binary change picture of cut area, 256*490. 126

67. Fertilizer 2 with Gaussian histogram, 420*350, red band. 128

68. Fertilizer 3 (Gaussian) registered with Fertilizer 2 (Gaussian) using nearest neighbor approximation, 420*350, red band. 128

69. Fertilizer 3 (Gaussian) registered with Fertilizer 2 (Gaussian) using bilinear approximation, 420*350, red band. 129

70. Fertilizer 3 (Gaussian) registered with Fertilizer 2 (Gaussian) using cubic convolution, 420*350, red band. 129

71. Binary change picture of Fertilizer area using nearest neighbor approximation, 420*350. 130

72. Binary change picture of Fertilizer area using bilinear approximation, 420*350. 130

73. Binary change picture of Fertilizer area using cubic convolution, 420*350. 131